UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Department of Pure and Applied Sciences

PH.D. PROGRAMME IN:

Research Methods in Science and Technology

CYCLE XXXVIII

IOT AND DIGITAL TECHNOLOGIES FOR QUALITY-ORIENTED CLEANING AND INTEGRATED SERVICES

ACADEMIC DISCIPLINE:

IINF-05/A – Information Processing Systems

Thesis written with the financial support of a scholarship co-financed by Ministerial Decree no. 352 of 9 April 2022, under the PNRR - financed by the European Union - NextGenerationEU - Mission 4 ‘Education and Research’, Component 2 ‘From Research to Business’ - Investment 3.3. ‘Introduction of innovative doctorates that respond to the innovation needs of companies and promote the recruitment of researchers from enterprises’ and by Papalini S.p.A.

Coordinator: Prof. Luca Lanci

Supervisor: Prof. Emanuele Lattanzi

Ph.D. student: Dott. Lorenzo Calisti

ACADEMIC YEAR
2024/2025

Abstract

Nowadays, people spend up to 90% of their time in enclosed spaces, so indoor hygiene is a critical yet often overlooked factor in public health. Despite this, the management of sanitization in built environments is still predominantly based on static schedules, visual inspections, and manual reporting, offering limited protection against invisible biological risks. This thesis addresses this gap by proposing integrated technologies for quality-oriented cleaning for intelligent hygiene management in smart buildings. The research is structured around three complementary pillars. The first tackles the problem of biological risk invisibility by introducing a virtual sensor. Instead of relying on slow and costly microbiological sampling, low-cost IoT sensors are used to continuously monitor environmental proxies such as CO_2, volatile organic compounds, temperature, humidity, and particulate matter. Through machine learning models, these quantities are mapped to airborne bacterial concentrations, transforming standard IoT nodes into virtual microbiological sensors. Experimental results demonstrate high predictive fidelity (R^2 up to 0.92) and sufficient accuracy to trigger preventive actions well before critical contamination thresholds, while deep learning–based power management strategies reduce wireless transmissions and extend battery life by up to 89%. The second research pillar is based on computer vision. Human occupancy is estimated by a virtual sensor extracting people counts, flows, and dwell times from video streams processed directly on edge devices. A lightweight tracking-by-detection pipeline, combined with a Dynamic Inference Power Manager, enables adaptive frame skipping based on scene dynamics. This approach achieves energy savings up to 36% without significantly degrading tracking accuracy, proving that visual sensing can be both sustainable and reliable for continuous deployment. The third pillar places human behavior at the center of hygiene dynamics through sensor-based Human Activity Recognition. A hierarchical architecture, named Lightweight Accurate Trigger, dramatically reduces wearable energy consumption by activating complex classifiers only when relevant motion patterns occur, achieving savings up to 95%. Generalized Zero-Shot Activity Recognition based on Siamese Neural Networks enables recognition of unseen activities without retraining, while Transformer-based models demonstrate robust signal reconstruction and real-time inference on resource-constrained wearable hardware. Overall, the results confirm that the proposed technologies are practical and scalable solutions, capable of combining high analytical accuracy with strict energy constraints. By integrating environmental, spatial, and behavioral intelligence, this thesis lays the foundation for predictive, adaptive, and verifiable hygiene management in future smart buildings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the contemporary era, the human species has effectively transitioned into what can be described as an "indoor generation". Statistical studies conducted in developed nations consistently indicate that individuals nowadays spend between 40% and 90% of their lives inside buildings [1]. These indoor environments include a wide range of spaces, such as residential homes, corporate offices, educational institutions, healthcare facilities, and transportation hubs. Given this profound shift in lifestyle, the built environment has evolved from a mere passive container of human activities into a primary determinant in shaping human health, comfort, and societal well-being. The management of indoor spaces is therefore critical on two distinct, but interconnected, levels. First, from an operational and aesthetic dimension. A facility that is dirty, neglected, or in disrepair negatively impacts the perceived value of the real estate and compromises the corporate image of the organizations housed within. Second, and more critical, from a biological dimension. The physical, biological, and chemical characteristics of these spaces exert a continuous influence on the physiological well-being of their occupants. Within this context, the management of hygiene, cleaning, and sanitization should be considered a core element, contributing directly to the reduction of health risks and the promotion of safe and healthy indoor conditions.

Historically, facility management has not approached cleaning as a rigorously defined scientific practice, but has instead regarded it mainly as an aesthetic task focused on maintaining visual order and decency. The prevailing goal has been to ensure that indoor environments appear clean and well-maintained, with performance judged through coarse, surface-level indicators. Floors are expected to be shiny, waste containers are regularly emptied, and surfaces are free from visible dust or debris. As a result, conventional cleaning practices emphasize visual appearance rather than quantifiable hygienic or microbiological effectiveness. Such approaches typically depend on manual procedures, generic cleaning agents, and rigid schedules intended to restore an acceptable visual baseline. The underlying assumption remains simple but flawed: if a space looks clean, it is presumed to be safe for human occupancy.

This visual-centric paradigm, while sufficient for maintaining appearances, fails to address the complex biological reality of enclosed spaces. This failure is encapsulated in the concept of Sick Building Syndrome (SBS), a medical condition recognized by the World

Health Organization (WHO) to describe various non-specific symptoms (mucosal irritation, skin reactions, and systemic fatigue) experienced by occupants of buildings with poor indoor air quality [2, 3]. A defining clinical feature is that these symptoms typically improve or resolve entirely once the individual leaves the building [4].

The causes and correlations of SBS are multiple and involve chemical, physical, and biological components. Chemical factors are among the most critical, particularly Volatile Organic Compounds (VOCs) and formaldehyde emitted from furniture and building materials [5]. Furthermore, carbon dioxide (CO_2) levels serve as a critical proxy for ventilation efficiency. Concentrations exceeding 1000 ppm are strongly correlated with the prevalence of symptoms [6]. Physical and biological factors, including dampness (which facilitates mold growth), inadequate lighting, occupational stress, and specific health histories, further exacerbate occupant distress [2]. As a result, SBS fundamentally reflects a complex interaction between the technical performance of buildings and the individual health profiles of occupants.

To understand the urgency of this issue, we must examine the mechanics of transmission of infections in the shared spaces. The primary vehicle is the fomite, an inanimate object that, once contaminated with infectious agents, can transfer disease to a new host [7]. In high-density environments such as airports, universities, or healthcare facilities, contamination occurs through two distinct yet overlapping pathways. The first is the localized contamination of Prime Points (PP). High-touch surfaces like door handles, elevator buttons, and handrails that are contaminated primarily through direct physical contact with unwashed hands [8]. The second, and often more insidious pathway, is the continuous respiratory saturation of the air. Beyond the immediate droplets produced by coughing or sneezing, the simple act of breathing by multiple occupants releases a constant stream of fine bioaerosols.

Over time, these suspended biological particles reach a saturation point in poorly ventilated areas and inevitably undergo a process of sedimentation, or bio-aerosol fallout. This passive accumulation means that infectious agents not only reside on high-touch nodes but gradually "rain down" onto all horizontal surfaces, including desks, equipment, and flooring, regardless of whether they have been directly touched.

The chain of infection is thus fueled by a constant environmental loading. Research indicates that the transfer rates of viruses between fingertips and surfaces are significant. On average, approximately 22% of the viral load present on a surface is picked up upon contact, with variations depending on the porosity of the material and the type of virus [9].

Once pathogens are transferred to the hands, the path to infection is extremely short due to the human habit of self-inoculation. Behavioral studies reveal that individuals touch their faces an average of 23 times per hour, creating a constant conduit for pathogens to reach the mucosal entry points of the eyes, nose, and mouth [10]. This hand-to-face trajectory is the primary driver for the spread of diverse, resistant agents, ranging from Norovirus and Influenza to MRSA and Coronaviruses. The challenge is exacerbated by the biological resilience of these pathogens. Rather than existing as isolated cells, many bacteria organize into biofilms: complex microbial architectures anchored to surfaces by a protective extracellular matrix [7]. Particularly in damp areas like sinks or drains, these biofilms act

as a shield against standard wiping and develop a high tolerance to chemical disinfectants, allowing colonies to persist for weeks or even months. Because these structures are invisible to the naked eye, surfaces that appear clean often function as active reservoirs that continuously re-contaminate the surrounding environment. Consequently, without a rigorous and scientifically grounded sanitization strategy that extends beyond mere visual inspection, these contact points become true reservoirs of infection, remaining capable of constantly re-contaminating the surrounding environment despite superficial cleaning efforts.

The consequences of failing to adequately control invisible biological risks are particularly severe in healthcare environments. Hospitals, which are designed to promote healing and recovery, can paradoxically become sites where Healthcare-Associated Infections (HAIs) originate, undermining their primary role and exposing patients to additional harm [11]. According to the Centers for Disease Control and Prevention (CDC), approximately one in twenty hospitalized patients in the United States acquires an infection linked to insufficient hygiene practices. Estimates from the 2011 report 721,800 HAI cases and nearly 75,000 related deaths each year [12, 13]. Similar trends are observed across Europe, where HAIs affect about 7.1% of inpatients and account for roughly 37,000 deaths annually [11]. In addition to the serious clinical impact, these infections place a significant financial strain on healthcare systems due to extended hospital stays, repeated treatments, and higher readmission rates, while also representing a substantial societal cost in terms of preventable morbidity and mortality.

The risk of infection is particularly critical in Intensive Care Units (ICUs), where infection rates can range from 51% in high-income countries to over 88% in low-income settings [11]. In these controlled environments, the microbial load is further increased by the circulation of multidrug-resistant bacteria and pathogens with high environmental persistence [14]. Gram-positive bacteria, for instance, can survive on surfaces for several months due to their resistant cell wall structure [15]. Consequently, high-touch surfaces, such as bed rails and medical equipment, act as stable reservoirs for pathogens [16]. This creates a significant risk of cross-contamination: patients admitted to rooms previously occupied by infected individuals face a higher probability of infection [11]. Furthermore, inadequate cleaning practices can unintentionally spread microorganisms across different areas, transforming maintenance tools into vectors for contamination rather than mitigation [17].

1.1 Conventional Cleaning Methods

Traditional cleaning techniques represent the building blocks of facility maintenance in man-made environments. Conceptually, these methods rely on consolidated processes of mechanical and chemical removal of contaminants. In this context, it is crucial to distinguish the semantic and operational differences between “cleaning” and “disinfection”, as these terms serve distinct functions. Cleaning refers to the removal of foreign material (i.e., soil and organic matter) from objects and surfaces. It is primarily a mechanical process enabled by water and detergents that suspend dirt and physically rinse away microbial loads. While cleaning reduces the number of pathogens, it does not necessarily kill them. Its critical role is to prepare the surface by removing organic matter that would otherwise act as a physical shield for microorganisms. Disinfection, on the other hand, is a process

that eliminates many or all pathogenic microorganisms present on inanimate objects. It is a chemical or thermal intervention targeted at biological inactivation. In professional protocols, the axiom is that "one cannot disinfect dirt". Without effective prior cleaning, disinfectants suffer from organic interference, significantly compromising the overall sanitization process. In professional protocols, disinfection cannot effectively occur without prior cleaning. The primary objective of professional cleaning is not necessarily the immediate killing of microorganisms, but rather their physical removal from surfaces [8]. In functional environments such as offices, schools, and hospitals, the operational goal is the transfer of pollutants from the interior to the exterior of the building, thereby reducing the microbial load to levels statistically considered safe for public health [17].

The execution of professional cleaning relies on rigorous systematic principles. A fundamental rule is directionality: operations must proceed from clean to dirty and from top to bottom to prevent cross-contamination between zones [18]. The efficacy of these processes is governed by Sinner's Circle, which balances four interdependent variables: (i) chemical action, (ii) mechanical force, (iii) temperature, and (iv) time. If the contribution of one factor is reduced, such as using lower temperatures, the intensity of the others must be increased to maintain the same hygienic standards [17].

To mitigate the risk of cross-contamination, the industry has adopted a universal color-coding system for equipment based on the biological risk of the zone in which they are used. Red and yellow tools are used in high-risk sanitary areas, such as toilets and washrooms, while blue and green are used in low-risk general areas and specialized zones like kitchens or sterile environments. [18]. Furthermore, the adoption of high-performance materials like microfiber textiles has replaced traditional cotton to better capture contaminants rather than spreading them.

Despite their widespread adoption, traditional methods are burdened by significant practical limitations. The most pervasive issue is the static nature of operations. Maintenance typically follows rigid calendars based on pre-determined shifts that are completely decoupled from the actual utilization of the premises. This disconnect leads to the "Phantom Cleaning" phenomenon, an inefficiency where resources are wasted sanitizing empty rooms while heavily trafficked areas degrade in hygiene quality between scheduled shifts [19]. This structural inefficiency is compounded by critical bottlenecks in time management. In high-pressure environments like hotels or corporate offices, operators work under strict time-per-room quotas that often make thorough cleaning impossible. Consequently, staff must resort to task rotation, performing deep cleaning, much like high dusting or scrubbing corners, only periodically. This accumulation of hidden dirt allows allergens and pollutants to build up over time. Additionally, physical obstructions play a major role. Approximately 70% of cleaning operators identify excessive furniture, desk clutter, and objects on floors as primary impediments that prevent access to critical surface areas [20].

From a biomechanical perspective, traditional cleaning constitutes an arduous, high-impact activity. The human body serves as the primary motor of the Sinner's Circle, resulting in significant physical strain. Statistics indicate that roughly 70% of professional cleaners perceive their work as physically stressful, a sentiment reflected in the high prevalence of Musculoskeletal Disorders (MSDs). The majority of operators report chronic pain localized in the lower back (57%), shoulders (56%), and neck (53%) [20]. These injuries

are the direct result of the repetitive movements and non-ergonomic postures inherent to manual tools. For instance, tasks such as dusting high surfaces or cleaning windows require working with unsupported arms and shoulders elevated above 60 degrees. Maintaining this posture for more than 10% of a work shift is a known risk factor for chronic shoulder impingement syndromes. Additional strain arises from the manual wringing of mops and cloths, which places stress on the wrists and forearms that often exceeds 30% of maximum voluntary muscular capacity. Tool design further contributes to the physical stress, as the use of mop handles of incorrect length forces operators into compensatory postures, either bending too low or reaching too wide. Over time, physical fatigue reduces operator precision, resulting in a gradual decline in cleaning performance as work shifts progress.

Ultimately, the most dangerous limitations of traditional methods lie in their inability to guarantee biological safety. As previously discussed, visual Inspection is a fallacious metric for hygiene; however, the procedural flaws extend beyond detection. Cross-contamination is an intrinsic risk of manual cleaning. If protocols are not strictly followed, the cleaning equipment itself becomes a vector. A mop head that is not changed frequently enough, or a bucket of cleaning solution that becomes contaminated with organic load, can spread microorganisms from a bathroom to a hallway, effectively distributing pathogens rather than removing them [21]. Furthermore, standard manual scrubbing is often insufficient to disrupt biofilms. These complex bacterial communities adhere strongly to surfaces and require specific mechanical agitation or enzymatic agents that are rarely part of a standard wipe-down routine.

Finally, the sector is affected by a pronounced lack of traceability. In conventional cleaning practices, compliance is typically documented through paper checklists completed and signed by operators, a process that is vulnerable to mistakes, missing information, and even deliberate manipulation. The absence of reliable, objective data makes it difficult for facility managers to verify hygiene levels or to identify recurring patterns over time [22]. In addition, the widespread use of harsh chemical products and synthetic fragrances to conceal odors frequently releases volatile organic compounds into indoor spaces, unintentionally degrading the very air quality that cleaning activities are intended to enhance.

1.2 Advancing Toward Digital Sanitization

The combined presence of operational inefficiencies, physical strain on workers, and unresolved hygiene risks indicates that the traditional cleaning model has reached its practical limits. These issues cannot be effectively addressed by simply increasing labor intensity or relying on stronger chemical agents. Such an approach would only exacerbate the biomechanical stress on operators and amplify the environmental impact associated with VOCs. Addressing these limitations instead requires a structural change in approach, moving away from a reactive and labor-intensive model toward a proactive, data-driven framework often referred to as "Cleaning 4.0". This new paradigm aims to overcome the disconnect between static cleaning schedules and the dynamic use of indoor spaces by integrating cleaning operations with Internet of Things (IoT) technologies and artificial intelligence (AI). The central objective is to replace assumptions of cleanliness based on visual inspections and

manual reporting with verifiable indicators derived from objective data.

By digitally modeling the state of buildings and how spaces are actually used, facility managers can move away from fixed, schedule-driven cleaning routines and adopt strategies that respond to real needs. In this context, cleaning activities are initiated by data signals rather than by predetermined calendars. Areas that are not used can be temporarily removed from cleaning plans, avoiding wasted effort, while spaces with unusually high occupancy can automatically raise alerts and require prompt action. This more selective approach helps limit cross-contamination by concentrating resources where the risk is greatest, and at the same time reduces the physical burden on cleaning personnel by removing repetitive tasks in areas where the risk remains low.

From an environmental perspective, the implications of this shift are profound. Traditional just-in-case cleaning results in the excessive and often unnecessary release of surfactants and biocides into the wastewater system. By implementing green protocols based on real-time monitoring and aligning cleaning frequency with actual usage, data-driven systems significantly reduce the chemical load imposed on the environment. Comparative studies highlight that such protocols can reduce the Global Warming Potential (GWP) of cleaning operations by 18.4%. This reduction is driven by a decrease in packaging waste, optimized water consumption, and the extended lifespan of consumables like microfiber pads [23]. The impact of this paradigm shift extends critically to energy consumption. With buildings responsible for approximately 40% of global energy usage, the management of heating, ventilation, and air conditioning (HVAC) systems is paramount [24]. Much of this expenditure is linked to inefficient ventilation strategies. The deployment of sensors to monitor CO_2 enables the dynamic regulation of mechanical ventilation. This approach resolves the historical dichotomy between air salubrity and energy conservation. While manual ventilation (i.e., opening windows) often results in massive thermal losses, AI-driven systems can precisely balance the air exchange rates required to prevent SBS while minimizing energy waste [25].

1.3 Open Research Frontiers

To bridge the gap between the invisible threat of pathogens and the macroscopic actions of cleaning staff, the building must evolve from a passive container into a cognitive, sensing entity. This transformation requires the deployment of a Cyber-Physical System (CPS) capable of continuously monitoring environmental parameters, quantifying human presence, and interpreting occupant behavior. This technological evolution rests upon three fundamental research pillars: (i) Environmental Monitoring, (ii) Occupancy Detection, and (iii) Human Activity Recognition.

The first fundamental step in mitigating biological risk is to make invisible threats visible through a pervasive sensory layer. Environmental monitoring architecture has moved beyond traditional, rigid systems to embrace Wireless Sensor Networks (WSN). By relying on the deployment of specialized hardware and low-cost sensors, these networks allow for the continuous acquisition of pollutant metrics and environmental parameters with high

spatial resolution. Data is transmitted in real-time to cloud servers or, in the event of network failure, stored locally on edge devices to ensure data integrity [26]. From an architectural perspective, WSNs offer superior scalability compared to traditional wired Building Management Systems (BMS) because they utilize communication protocols specifically designed for IoT applications. These protocols ensure long-range connectivity and low power consumption, which is ideal for covering vast areas like hospital districts or university campuses, allowing sensors to operate for years on a single battery [27]. Furthermore, the implementation barrier for these networks has been significantly lowered by the integration of open-source hardware, enabling the deployment of capillary monitoring systems using microcontrollers with a cost per node as low as $30 [3].

An advanced monitoring system makes use of this infrastructure to observe a broad spectrum of indicators that define biological salubrity. Central to this analysis is carbon dioxide (CO_2), utilized in facility management as a primary metric for assessing ventilation efficiency and occupant density. High levels signal a potential accumulation of bioaerosols, statistically correlating with an increased risk of pathogen transmission [25]. This atmospheric analysis is also linked to the monitoring of particulate matter (i.e., $PM_{2.5}$ and PM_{10}), as these microscopic particles serve as physical vectors capable of transporting microorganisms deep into the respiratory system [15]. Concurrently, the system tracks VOCs and Formaldehyde via metal-oxide sensors to detect chemical pollutants originating from construction materials or, paradoxically, from the overuse of aggressive cleaning agents. To achieve a more direct assessment of biological risk, these setups can detect Bioaerosols, such as E. Coli or S. Epidermidis, through optical sensors or automated ATP bioluminescence. In specific contexts like restrooms, they employ gas sensors to identify Ammonia peaks, triggering immediate cleaning interventions before odors become perceptible [19].

Despite these operational benefits, moving toward a data-rich environment presents a number of nontrivial technical issues. Sensor calibration, in particular, is still a largely unresolved problem: low-cost sensing devices are well known to exhibit signal drift over time and to be affected by cross-sensitivity phenomena. As an example, sensors designed to detect ammonia may also respond to other organic amines, requiring advanced software-based compensation techniques to preserve measurement reliability [28]. In parallel, the management of the large data volumes produced by thousands of sensors operating at high sampling rates places a considerable strain on the underlying infrastructure. The transmission of raw data at this scale can easily result in network congestion and increased latency. For this reason, contemporary system architectures increasingly rely on Edge Computing, where data is locally processed and pre-filtered at the gateway. In this way, only relevant information and critical alerts are forwarded to the cloud, reducing bandwidth usage and enabling timely reactions to potential hygiene risks [27].

While environmental sensors are effective in capturing the chemical and physical conditions of indoor spaces, they provide limited insight into how these spaces are actually used. To address this limitation, the second pillar of the Cleaning 4.0 framework focuses on Occupancy Detection. The objective is to quantify how many people access a given area and to characterize their movement patterns over time. This information enables a shift from time-based cleaning schedules to usage-based interventions, in which sanitation

activities are triggered after a defined level of use, such as cleaning a restroom after a specific number of accesses. In this way, high-traffic areas can be treated before they become critical points for contamination.

Several sensing technologies have traditionally been employed for occupancy monitoring, each with notable limitations. Passive Infrared (PIR) sensors detect human presence based on body heat and are widely adopted due to their low cost and energy efficiency. However, they typically provide only binary information (presence or absence) and are unable to accurately count occupants, particularly when individuals remain stationary or move in groups [29]. Infrared (IR) beam-break sensors, commonly installed at entrances, estimate occupancy by counting whether a light beam is interrupted. These systems are prone to occlusion errors, often miscounting multiple people walking side by side as a single individual. Also, they cannot reliably distinguish between entry and exit events without more complex dual-beam configurations [19]. More specialized solutions, such as RFID-based systems used in healthcare settings to monitor staff movement and hand hygiene, are also limited, as they do not capture the presence of visitors or patients who lack the required tags [11].

To overcome these limitations, modern facility management increasingly relies on Computer Vision (CV) techniques. By using cameras as high-resolution optical sensors, buildings can apply Deep Learning algorithms to analyze video streams in real time. The first stage of this process involves people detection, where models such as YOLO (You Only Look Once) or SSD (Single Shot MultiBox Detector) identify and localize human figures within each frame. Unlike traditional infrared sensors, CV can distinguish people from objects such as cleaning carts or service robots, reducing false detections. It also supports the simultaneous detection of multiple individuals, identifying not only their presence but also the overall density and distribution of a crowd [30]. However, detection alone provides only a static snapshot. To capture how spaces are used over time, people tracking algorithms are required. Methods such as DeepSORT or Intersection over Union (IoU) assign unique and persistent identifiers to each detected person, following their trajectories across consecutive frames. This transforms camera systems into tools for advanced spatial analysis, making it possible to calculate real-time occupancy by monitoring crossings of virtual boundaries and to reconstruct movement paths within a space. By identifying which areas are most frequently accessed and which surfaces are likely to be touched, tracking data supports highly targeted hygiene interventions.

When aggregated over longer periods, tracking information can be used to generate occupancy heatmaps that highlight the most heavily trafficked zones. Rather than cleaning an entire 500 m^2 lobby, staff or autonomous cleaning systems can focus on the limited portion of the area that accounts for the majority of foot traffic [12]. In addition, this data enables forms of social signal processing, such as detecting stationary crowds at reception desks or elevators and distinguishing between cohesive groups and incidental contacts. These capabilities proved particularly valuable during the COVID-19 pandemic for monitoring distancing and personal protective equipment (PPE) compliance, and they remain relevant for activating localized air treatment or ventilation adjustments when the risk of bioaerosol accumulation increases [19].

The adoption of camera-based sensing systems inevitably raises concerns related to

privacy protection and regulatory compliance, particularly within the framework of data protection regulations such as the European General Data Protection Regulation (GDPR). Under these regulations, video data that can directly or indirectly identify individuals is classified as personal data and is therefore subject to strict requirements regarding collection, processing, storage, and access. For this reason, the use of conventional video surveillance approaches is often incompatible with large-scale occupancy monitoring in public or semi-public indoor environments. Cleaning 4.0 addresses these constraints through the use of edge artificial intelligence (Edge AI) and a privacy-by-design methodology. Video streams are processed locally, either directly on the camera or on a nearby edge gateway, eliminating the need to transmit raw images to centralized servers or cloud infrastructures [31]. The analysis is performed in real time, and only aggregated non-identifying metadata, such as occupancy counts, movement vectors, or density indicators, are retained. The original video frames are handled exclusively in volatile memory and are immediately discarded after processing, ensuring that no visual records are stored. This architectural choice significantly reduces the risk of unauthorized access or data misuse and supports compliance with key GDPR principles, including data minimization, purpose limitation, and storage limitation. By ensuring that no biometric or identifiable information is collected or preserved, the system effectively reframes cameras as quantitative sensors rather than surveillance devices. In environments with particularly strict privacy requirements, visual sensing can be further reduced or avoided altogether through the integration of radar-based technologies. These systems are capable of detecting presence, movement, and crowding without capturing images, offering an additional layer of compliance while still providing the granular occupancy data required for data-driven and hygienically effective facility management [30].

The third and final layer of the Cleaning 4.0 framework is Human Activity Recognition (HAR). While environmental sensing identifies when cleaning may be required, and occupancy detection determines where interventions should take place, HAR focuses on understanding what people are actually doing within indoor spaces. This distinction is essential because humans are the primary vectors of biological contamination, and the mere presence of occupants does not fully explain hygienic degradation. Instead, it is human behavior that largely determines the level and distribution of biological risk [28].

Recent advances in HAR concentrate on identifying behaviors that are directly associated with pathogen transmission, such as coughing, sneezing, or frequent interaction with high-risk surfaces. Detecting these events in real time makes it possible to initiate highly targeted sanitation actions, addressing contamination at its source rather than applying uniform and generalized cleaning procedures. The importance of activity-aware monitoring becomes particularly evident when considering indoor pollutant dynamics. In relatively static environments such as offices or classrooms, where occupants remain seated for long periods, pollutant accumulation is mainly driven by metabolic processes, including carbon dioxide emissions and the passive release of bioaerosols [32]. In contrast, dynamic activities significantly increase hygienic risk, as intense movement can resuspend particles previously deposited on floors and surfaces, leading to a rapid deterioration of indoor air quality [6]. By recognizing these behavioral patterns, cleaning strategies can shift toward a usage-based maintenance model, in which cleaning frequency and intensity are adjusted

only when observed activities indicate that safety thresholds may be exceeded.

To monitor human activities while preserving individual privacy, Cleaning 4.0 increasingly relies on wearable technologies rather than vision-based systems. Devices such as smartwatches and smart bands are equipped with Inertial Measurement Units (IMUs) that capture motion-related signals, including changes in acceleration and orientation of body segments. In healthcare environments, these sensors are often embedded in dedicated wristbands designed to monitor critical actions, particularly hand hygiene practices. By analyzing movement patterns, these systems can assess whether handwashing procedures comply with the multi-step protocols recommended by the WHO [33, 11].

The transformation of raw inertial signals into meaningful activity labels is enabled by advanced machine learning models. Deep learning architectures designed for time-series analysis can capture the temporal structure of human motion, allowing the system to distinguish between similar actions and recognize complex sequences of movements. More recently, these capabilities have been extended through TinyML approaches, which allow inference to be performed directly on the wearable device itself. Processing data locally ensures immediate responsiveness while maintaining a high level of privacy, as sensitive movement information never leaves the device [27]. Beyond passive monitoring, HAR-enabled wearables support active behavioral reinforcement. They can provide real-time feedback, for example, through haptic signals, to remind healthcare workers to sanitize before patient contact or to indicate that a handwashing procedure was too short or incorrectly executed. In combination with ambient sensing, wearables can also generate contextual alerts, notifying users when they may have been exposed to contamination after interacting with a surface identified as high risk.

The convergence of environmental sensing, occupancy detection, and human activity recognition represents a clear transition from reactive cleaning practices to a closed-loop hygiene management system. In this integrated model, risks are identified in real time rather than inferred a posteriori, allowing sanitation actions that are both precise and efficient. By continuously linking environmental conditions, space utilization, and human behavior, cleaning interventions can be driven by actual risk levels rather than static assumptions. Despite the extensive literature on each of these technologies, a relevant gap persists in current research. IoT sensing, CV, and HAR are often investigated as separate technological domains, with limited attention to their joint operation under real-world constraints. However, a truly smart system requires these components to cooperate dynamically. For example, increased occupancy should trigger more frequent environmental sampling, while prolonged vacancy should allow computationally intensive analytics to be suspended or significantly downscaled.

The central objective of this thesis is to analyze and mitigate the energy consumption associated with advanced sensing and inference technologies used in intelligent facility management. IoT sensing infrastructures, vision-based systems, and HAR models can be highly energy-intensive, especially when continuous monitoring is implemented through deep learning techniques. This work investigates how energy consumption can be reduced without sacrificing the reliability and accuracy required by hygiene-critical applications.

To address these gaps, this thesis identifies and explores four fundamental Research Questions (RQs):

- **RQ1: Biological Risk Monitoring via Low-Cost Sensors.** Is it possible to transform standard, low-cost IoT nodes into reliable "microbiological sentinels" by using DL models to estimate airborne bacterial load from environmental proxies, thus bypassing expensive manual sampling?
- **RQ2: Privacy-Preserving Occupancy Estimation.** How can Computer Vision be integrated into a Cleaning 4.0 framework to extract real-time occupancy metrics and data flows, while ensuring total data privacy through Edge computing and maintaining energy sustainability on limited hardware?
- **RQ3: Traceability of Human Behavior.** How can wearable inertial sensors and HAR bridge the traceability gap in manual cleaning, allowing not only for the validation of hygiene protocols but also for quantifying the contamination load generated by occupant activity?
- **RQ4: Analytical Accuracy vs. Energy Constraints.** What algorithmic strategies, such as PBDC, hierarchical triggers, or DIPM, are necessary to deploy complex AI models on embedded and wearable devices without compromising the battery autonomy required for continuous monitoring?

To this end, the thesis examines energy-aware design strategies specific to each technological domain, including adaptive sampling mechanisms for environmental sensing, duty-cycling, and selective activation schemes for occupancy and vision-based detection, and lightweight inference pipelines based on TinyML for human activity recognition. Rather than being addressed as a secondary optimization goal, energy efficiency is treated as a primary design constraint, directly influencing both system architectures and algorithmic choices throughout the proposed solutions.

The remainder of this thesis is structured as such. Chapter 2 provides a comprehensive review of the state of the art in environmental sensing, occupancy detection, and human activity recognition, with particular emphasis on existing approaches to energy efficiency, edge computing, and privacy preservation. The core of the thesis is then organized around three main research streams, each discussed in a dedicated chapter: environmental monitoring (Chapter 3), occupancy detection (Chapter 4), and recognition of human activity (Chapter 5). Each of these chapters follows a common and coherent structure, comprising a background section, a detailed description of the proposed methodology, the experimental setup, and an in-depth discussion of the results obtained within the specific domain. Finally, Chapter 6 summarizes the main findings of the thesis and outlines future research directions, with a focus on the adoption of Cleaning 4.0 technologies in public and healthcare settings.

Chapter 2

Related Works

Professional sanitization and cleaning have evolved far beyond simple operational tasks. Today, they represent a structural component of integrated services within complex buildings, having a direct impact on public health, indoor environmental quality, and how users perceive the safety of a facility. In critical contexts such as hospitals, schools, universities, and large public hubs, cleaning must be interpreted as a continuous, measurable, and critical process. It is strictly connected to the security, sustainability, and management efficiency of our built environments [34, 19]. Despite this central role, the traditional management of cleaning activities still faces numerous structural challenges. The models currently adopted in the industry rely predominantly on static and pre-scheduled programs. These schedules are often defined independently of the real hygienic conditions or the actual utilization of the spaces. Furthermore, quality control typically depends on manual inspections and visual checks. This approach is subject to human bias and lacks objective traceability, making it difficult to guarantee uniform quality standards [35, 36].

A major limitation of this traditional approach is the difficulty in objectively evaluating what "clean" really means. Fundamental parameters such as microbiological contamination or invisible hygienic degradation are not perceivable by the human eye. Consequently, facility managers are often forced to operate with a reactive logic rather than a preventive one [37]. Without structured data, it becomes nearly impossible to optimize processes, avoid resource waste, or prove compliance with strict health regulations. This issue is compounded by the dynamic nature of modern buildings, where variable flows of people and diverse activities create a misalignment between static cleaning schedules and real-time needs [38].

In response to these challenges, modern information technologies offer a significant paradigm shift. The integration of Internet of Things (IoT) sensors allows us to transform cleaning from an experience-based activity into a data-driven process. By continuously collecting data on parameters such as indoor air quality (IAQ), temperature, and humidity, we can move beyond static schedules to enable adaptive sanitization strategies. In this scenario, the building becomes a cyber-physical system capable of perceiving its own state and supporting informed decisions through Artificial Intelligence and Machine Learning [36, 39]. However, monitoring environmental parameters is only part of the solution. To truly optimize sanitization, we must understand how spaces are inhabited and how people

interact with them. This is where computer vision plays a crucial role. Advanced vision systems allow us to analyze occupancy levels and movement flows in real-time. By detecting which areas are crowded and which are empty, facility managers can direct cleaning resources exactly where they are needed, improving efficiency and reducing unnecessary labor. Complementing the analysis of spaces is the analysis of human behavior, achieved through human activity recognition (HAR). Since human interaction with surfaces is the primary vector for contagion, monitoring specific behaviors is essential for a complete sanitization strategy. By using wearable sensors to detect and evaluate hand-washing gestures, we can ensure compliance with hygiene protocols, effectively breaking the chain of contamination between surfaces and occupants [40].

In this chapter, cleaning is approached from a systemic perspective, not as a standalone task but as a service embedded within the broader smart building ecosystem. The chapter reviews the state of the art along four interconnected technological directions. First, it surveys current cleaning and sanitization practices, highlighting the growing emphasis on objective, quantifiable indicators of quality. Second, it examines IoT-based solutions for indoor environmental monitoring, with particular focus on continuous sensing and energy efficiency. Third, it investigates the use of computer vision techniques for occupancy detection and space utilization analysis in support of quality-driven service planning. Finally, it addresses Human Activity Recognition, concentrating on hand hygiene monitoring, energy consumption issues, and deployment on resource-constrained devices. Taken together, these perspectives describe the transition of cleaning activities from manual, reactive procedures to data-driven and predictive processes aimed at improving occupants' well-being.

2.1 Sanitization and Professional Cleaning

Contemporary research on the management of human-made environments is undergoing a significant transformation. This shift is driven by the growing need for higher hygiene and health standards in complex public settings. The cleaning and maintenance of high-traffic locations such as hospitals, hotels, and shopping centers represent a critical public health challenge. Traditional approaches often prove ineffective, costly, and difficult to track [19]. High occupancy density and continuous user turnover make rigid time-based intervention models inadequate. The functional complexity of these environments also renders subjective hygiene assessments unreliable.

This issue is even more significant within the healthcare sector. Healthcare-Associated Infections (HAI) represent a systemic threat. Approximately one in twenty patients contracts an infection due to hygiene deficiencies. This has a direct impact on both morbidity and overall healthcare costs [12]. Numerous studies show that these infections are frequently associated with contaminated surfaces and devices. Items such as bed rails, handrails, handles, and buttons act as microbial reservoirs. Without effective and repeated removal procedures, such surfaces can host pathogenic microorganisms for extended periods. These pathogens can persist for several months and contribute to a sustained infection risk [11, 16].

At the same time, the evolution of building and facility management systems reflects a progressive paradigm shift occurring in this field. Cleaning and sanitization activities

are no longer viewed as mere auxiliary services. They are now considered structural components of public health strategies and organizational safety. These activities also affect the perceived quality of environments [41]. Although managing the living and working environment has historically been linked to human survival, only in the last two centuries has systematic sanitation been formalized as a scientific tool for pathogen control. This process transformed empirical practices into the rigorous methodologies used by modern sanitarians. These methods are based on standard procedures, performance metrics, and verifiable protocols [17].

While traditional cleaning relies on the rigorous application of Sinner's Circle and color-coding protocols, recent literature highlights the inherent limitations of these manual systems in complex environments like hospitals or shopping malls. The primary challenge is not the efficacy of the protocols themselves, but their lack of objective verification [8]. The mechanical removal of organic matter and the balancing of chemical-thermal variables remain the gold standard for hygiene. However, in non-standardized operational practices, the critical factor of contact time is frequently neglected, leading to suboptimal sanitization [17]. Recent studies emphasize that even with strict "top-to-bottom" and "clean-to-dirty" sequences, the risk of cross-contamination remains high due to human error and the difficulty of monitoring adherence in real-time.

Beyond the procedural limits of manual cleaning, conventional methodologies suffer from a structural lack of flexibility. Currently, most professional cleaning is based on static scheduling that does not account for actual room usage or real-time hygienic conditions [19]. In high-traffic environments like shopping centers or hotels, this rigidity leads to a double failure: a waste of resources during low-occupancy periods and inadequate sanitation during peak hours [42]. This "blind" management is further complicated by three critical factors that highlight the need for digital transformation. First, the documentation tracking the work's progress is still largely paper-based, making it prone to errors or untruthful entries. This prevents managers from objectively verifying if quality standards, such as the correct contact time for disinfectants, were actually met [22, 13]. Second, traditional cleaning remains a high-strain activity. Many operators suffer from musculoskeletal disorders due to repetitive, non-optimized movements [20]. Furthermore, the lack of real-time waste monitoring increases the risk of accidental exposure to hazardous materials [17]. Lastly, the intensive, often unnecessary, use of water and chemical detergents in static protocols has a significant ecological footprint. Research indicates that transitioning to "green" and data-driven protocols can substantially reduce CO_2 emissions without compromising hygiene levels [23].

2.2 Towards Data-Driven Hygiene

These critical issues constitute the primary driver for adopting solutions based on Internet of Things (IoT) systems and artificial intelligence (AI). The objective of this transition is not the mere automation of existing activities, but transforming cleaning into an intelligent and adaptive service oriented toward measurable quality indicators [27]. Conventional models relying on fixed schedules and manual records are unable to respond effectively to the variable dynamics of modern buildings [19, 42]. Conversely, the integration of

IoT sensors enables a reactive and data-driven management where decisions are based on objective data collected in real time [13].

A clear example of this change is represented by smart restrooms. Ammonia (NH_3) sensors and access monitoring systems are used in these environments to allow cleaning interventions to be activated only when critical thresholds are reached, optimizing resources based on the actual level of degradation [19]. This logic extends to the management of large indoor spaces through wireless sensor networks (WSN). Low-power communication protocols such as [43] and [44] enable continuous and scalable monitoring, improving environmental comfort and reducing energy consumption by 15% to 40% [45, 46].

The digitalization of cleaning workflows introduces the central theme of operational traceability. Within this context, the adoption of Blockchain technology enables the immutable recording of sanitization activities, creating chains of verifiable evidence that support quality assurance processes and auditing activities [13]. Integration with Radio Frequency Identification (RFID) systems allows each intervention to be associated with a specific operator, a defined area, and a precise time window, which reduces the risk of data errors and manipulation [13].

The integration of Low-Cost Sensors (LCS) for particulate matter and devices for monitoring volatile organic compounds (VOC) and harmful gases within IoT platforms enables the real-time identification of pollution hotspots in offices and hospitals [26, 15, 47]. Through Cloud Computing platforms and machine learning algorithms, it is now possible to train these devices to differentiate between non-biological aerosols and bioaerosols such as bacteria and viruses, offering a rapid tool for real-time infection risk assessment [15, 48]. Deep Learning architectures, including Autoencoders and Long Short-Term Memory (LSTM) networks, are particularly suitable for the analysis of multivariate time series, as they allow for the identification of anomalous deviations in environmental parameters before these result in actual risk conditions [49].

This information flows find a natural placement in the evolution from Building Information Models (BIM) toward the Digital Twin (DT) paradigm. While traditional BIM provides a primarily static representation of the building focused on geometric, structural, and system aspects, the DT introduces a dynamic and temporal dimension, allowing the virtual model to be associated with the real operational state of the physical asset during the entire operation and maintenance (O&M) phase [50]. In this context, the DT becomes the convergence point for information originating from IoT systems distributed throughout the building. Environmental sensors, energy monitoring devices, occupancy counters, and cleaning activity tracking systems feed the digital twin in real time, transforming it into an advanced decision-making platform for facility management [51].

This architecture allows facility managers to interact with the building through dedicated mobile applications, receiving push notifications, SMS, or emails in real time or when critical thresholds are exceeded. The ability to graphically consult parameter history facilitates rapid decisions, such as activating ventilation or turning off polluting equipment, and permits immediate automated sanitization interventions [3, 24, 2]. In this scenario, cleaning is no longer an isolated activity but becomes an integral part of an intelligent ecosystem where ventilation, disinfection, and energy management cooperate to ensure a healthy, productive, and certifiable environment. This approach enables the transition

from reactive or corrective maintenance to predictive maintenance, which is based on the real state of systems and actual environmental conditions [52, 49].

Validating the effectiveness of these interventions requires reliable and standardized quality control tools. In this regard, the adoption of structured cleaning bundles, which include systematic staff training and the targeted cleaning of high-touch points, represents a consolidated best practice. The use of UV-C fluorescent markers allows for an objective and repeatable verification of the cleaning of critical surfaces, overcoming the limitations of traditional visual inspections [16]. Clinical evidence demonstrates that the rigorous application of such protocols is associated with a significant reduction in the incidence of healthcare-associated infections caused by multi-resistant pathogens, such as Vancomycin-resistant Enterococci (VRE) [16]. Finally, the technological modernization of cleaning and building management services has significant implications for environmental sustainability. Recent studies indicate that the adoption of IoT systems for the joint optimization of HVAC, lighting, and occupancy can lead to overall energy consumption reductions of up to 30% without compromising comfort and indoor environmental quality [53, 25].

2.3 Occupants Tracking and Behavior

Human presence represents the main vector of contamination in buildings, as occupants contribute to the emission of CO_2, respiratory aerosols, bioaerosols, and anthropogenic VOCs. Occupants also determine the hygienic degradation of surfaces through direct and indirect contact. In this sense, accurate knowledge of occupancy levels constitutes a central factor for planning cleaning, sanitization, and ventilation interventions of indoor environments [54]. Literature highlights how technologies for occupancy detection cover a very broad spectrum, ranging from passive environmental sensors to advanced computer vision systems. Each approach presents specific trade-offs in terms of accuracy, latency, cost, scalability, intrusiveness, and social acceptance. This makes technology selection highly dependent on the application context, particularly in sensitive environments such as hospitals, schools, and public offices [55, 56].

Environmental sensors are among the less invasive solutions, being easy to integrate into existing building automation systems. In particular, CO_2 concentration is widely used as a proxy for human presence, since occupants are the primary source of carbon dioxide in indoor environments [56]. However, the use of CO_2 to estimate occupancy is affected by an intrinsic latency related to gas accumulation, diffusion, and removal processes. This makes it less suitable for contexts characterized by intermittent occupancy or high air exchange rates [57]. To mitigate these limitations, models based on mass balance equations and probabilistic approaches have been proposed to infer occupancy from the temporal evolution of CO_2. However, their transferability between different buildings remains limited, as these models depend heavily on the characteristics of the building envelope, air volumes, and the ventilation strategies adopted [54, 56]. A more robust approach is represented by sensor fusion, which combines CO_2 with temperature, relative humidity, and VOC concentration. Although these parameters show a weak correlation when considered individually, their integration into multivariate models allows for a significant improvement in the accuracy of estimates [57, 58]. In particular, studies based on Linear Discriminant

Analysis (LDA) show that the combination of temperature and artificial lighting allows for effective discrimination of occupancy status, reaching accuracies greater than 95% [58].

Alongside environmental sensors, motion detection technologies such as passive infrared (PIR) sensors and ultrasounds offer a more immediate response to presence events [59]. However, such solutions generally provide binary information regarding the presence or absence of people and present significant limitations from a sanitization perspective. PIR sensors do not detect static occupants, while ultrasonic sensors can generate false positives in environments with complex geometries [60, 61]. To overcome these critical issues, doorway counting systems based on active infrared have been developed. These devices allow for the bidirectional counting of entries and exits with accuracies up to 97% while ensuring the complete anonymization of users [62]. This approach is particularly relevant for healthcare and school facility management, where the cumulative number of passages constitutes a direct indicator of potential hygienic degradation and can be used as a trigger for cleaning interventions based on the actual use of spaces.

A further line of research concerns the use of opportunistic data, also known as implicit sensing, such as counting the number of devices connected to Wi-Fi or Bluetooth networks. These approaches show a strong correlation with the real number of occupants, with coefficients of determination R^2 ranging between 0.80 and 0.83 [57, 56]. Although highly scalable and low-cost, these systems introduce critical issues related to privacy, personal data management, and the variability in the number of devices per individual [63]. Complementarily, monitoring electrical loads through smart meters provides an indirect estimate of occupancy. The analysis of aggregated loads, such as plug and light loads, has proven highly effective, especially in environments with regular usage patterns [57, 56].

Computer vision technologies provide a very rich source of information, as they are able not only to count the number of occupants but also to analyze activities, spatial distribution, and movement flows within indoor environments [55]. Their integration into IoT systems has opened new opportunities for intelligent environment management, where tracking people and objects supports not only security functions but also the optimization of cleaning, sanitization, and integrated services. Data such as occupancy levels, dwell time in specific areas, movement patterns, and space usage are key inputs for planning quality-oriented cleaning actions, improving resource allocation, and reducing health risks. At the same time, the adoption of vision-based systems faces significant challenges related to social acceptance, especially in sensitive contexts such as healthcare. Post-pandemic studies, including the SAFE PLACE project, show a clear preference for automated and non-intrusive sanitization solutions over visual monitoring approaches perceived as invasive [31]. From a technical perspective, traditional architectures based on streaming video to centralized servers raise concerns about latency and privacy [64]. Running object detection and tracking algorithms directly at the edge helps address these issues by reducing latency, enhancing privacy protection, and improving the scalability of such systems.

The transition to the Edge is supported by the availability of hardware accelerators such as Graphical Processing Units (GPU) and Tensor Processing Units (TPU), which allow the execution of intensive tasks like Object Detection (OD). Although literature has extensively investigated the feasibility of detection on such accelerators in terms of accuracy and energy efficiency [65, 66, 67, 68], the tracking phase has often been neglected. In

continuous monitoring scenarios, tracking is instead essential to provide temporal continuity to occupancy data. A first attempt to balance tracking and energy savings can be found in the adoption of lightweight algorithms that adapt camera idle periods based on object speed [69]. However, these approaches show limits in crowded environments, which are typical of public buildings or hospitals. Other efforts have focused on distributing the computational load between IoT nodes and edge servers to respect latency constraints, as seen with YOLOv2-tiny networks partitioned across different devices [70], or the use of background-aware correlation filters to improve system robustness [71].

Other works have addressed multi-object tracking on higher-performance edge hardware. Blanco-Filgueira et al. proposed a real-time multi-object visual tracking system on NVIDIA Jetson TX2 [72]. In this case, the authors provide experimental measures of energy consumption as the number of tracked objects and board operating modes vary, highlighting the strong link between computational load and power consumption. Complementarily, Inoue et al. proposed energy-aware tracking algorithms based on frame rate adaptation according to object speed [73, 74]. However, these approaches were primarily evaluated through simulations without a real hardware implementation. A relevant contribution toward the complete execution of detection and tracking on highly constrained devices is represented by Paissan et al. [75]. The authors propose a lightweight backbone based on MobileNet, combined with YOLOv2 as a detector and SORT as a tracker, implemented on an STM32H743 microcontroller.

From a hardware perspective, several studies have evaluated the energy efficiency of edge platforms like NVIDIA Jetson Nano and Google Coral AI. Puchtler et al. [66] and Baller et al. [76] compare different accelerators and software frameworks in terms of accuracy, latency, and energy consumption for detection. More recent studies analyze the impact of different YOLO versions on NVIDIA boards [77, 78]. However, in most cases, the energy contribution of tracking is not considered. Regarding tracking specifically, Danish et al. [79] analyze detection and tracking pipelines based on DeepSORT, measuring energy consumption on Raspberry Pi 4B and Google Coral TPU. Bhatti et al. [80] measure the consumption of the Jetson Nano in vehicle and pedestrian detection and tracking scenarios. Despite this, a systematic characterization of the trade-off between tracking accuracy, algorithm choice, and energy consumption is still missing, especially in applications oriented toward indoor environment management.

In the context of indoor environment management, people tracking is particularly relevant for applications such as people counting, flow analysis, and stay-time estimation, which are all fundamental for planning targeted cleaning interventions. Some works propose systems for detecting, tracking, and counting people based on YOLOv5 and IOU/V-IOU trackers, but without an energy evaluation [81]. Other authors compare different tracking algorithms such as SORT, Deep SORT, and Hungarian in sports scenarios, focusing on accuracy metrics but not on efficiency [82].

A complementary approach to extend device autonomy is the use of adaptive frame rate techniques. While initially developed to improve responsiveness in facial recognition or real-time detection [83, 84], these techniques have been applied in the energy domain to reduce consumption during video encoding and streaming [85, 86]. However, the literature highlights a lack of systematic characterization regarding the trade-off between tracking

accuracy and consumed energy, especially for algorithms based on Intersection-over-Union (IOU). Although appreciated for their simplicity on edge devices and UAVs [87, 88], IOU trackers require extremely fine parameter calibration. The influence of this calibration on energy efficiency has not yet been fully investigated despite numerous attempts to improve robustness or integrate it with more complex approaches [89, 90, 91, 92, 93, 94].

Siamese Neural Networks (SNN) represent an advanced frontier for tracking due to their intrinsic ability to compare similarity between input pairs through branches with shared weights [95, 96]. This characteristic makes them ideal for single-object and multi-object tracking, even in combination with traditional detection-based approaches [97]. The implementation of SNN on embedded hardware has seen the use of platforms like Jetson Nano and Coral AI to analyze parameters such as latency and throughput [98], leading to the optimization of complex models for specific targets [99, 100]. Despite the ability of Siamese networks to guarantee tracking continuity even across different domains [101], research on the energy efficiency of these architectures is still in its infancy. Rare studies that consider energy consumption, for example, in facial recognition systems [102], fail to isolate the computational cost of the Siamese component, leaving the potential of SNNs on TPU-based devices or microcontrollers unexplored from an energy sustainability perspective.

2.4 Human Activity Recognition for Behavioral Monitoring

Hand hygiene represents a fundamental element connecting surface cleaning, indoor air quality, and human behavior within built spaces. Literature agrees in considering hand washing and alcohol-based rubbing as the single most effective measure to reduce pathogen transmission and the incidence of healthcare-associated infections (HAI) in both healthcare settings and high-density collective environments [103, 104, 34]. From this perspective, hand hygiene should not be treated as an isolated or exclusively clinical practice but must be interpreted as an integral component of environmental sanitization strategies capable of breaking the contamination chain.

Hand hygiene is recognized by the World Health Organization (WHO) as the first line of defense against the spread of nosocomial infections and viral pandemics. Many pathogens, including SARS-CoV-2, are transmitted through contact, especially when contaminated hands touch the mouth, nose, or eyes [105]. For this reason, the WHO has defined structured hand hygiene procedures, including hand-rubbing with alcohol-based solutions, consisting of 8 steps performed for 20–30 seconds, and hand-washing with soap and water, consisting of 11 steps performed for 40–60 seconds. Compliance with these protocols has been assessed through direct observation by trained personnel, long considered the gold standard for monitoring practices such as the "5 Moments for Hand Hygiene" [103, 106]. However, this approach has important limitations, including high human resource costs, subjective evaluations, and poor scalability. In addition, compliance often improves only temporarily when individuals are aware of being observed, reducing the long-term effectiveness of manual monitoring [103, 107, 106]. These issues are further amplified by the intrinsic complexity of hand hygiene procedures, which frequently leads to low adherence in everyday practice, with users performing unstructured actions and neglecting the specific

movements recommended by official guidelines.

Recent research has introduced electronic monitoring systems based on IoT that allow for the automatic and continuous detection of hygiene events [103, 34]. The use of badges, wearable devices, and smart dispensers enables the correlation of hand washing with spatial and temporal contexts. Recent work in this field has shown that common commercial smartwatches can be transformed into proactive monitoring tools using deep learning algorithms to recognize even unstructured hand washing movements with high accuracy [40, 64]. These systems are particularly relevant because they can provide immediate feedback, such as vibrotactile or visual signals, notifying the operator if hygiene is not performed before contact with patients or critical surfaces [103, 104].

The effectiveness of various solutions varies significantly depending on the application context. In Emergency Intensive Care Units (EICU), the introduction of IoT systems has shown a significant improvement in adherence among doctors and nurses, while less evident results were observed for cleaning staff [103]. In assisted living facilities and nursing homes, the problem takes on even more complex connotations where the rigid application of hospital protocols risks compromising the quality of life of guests. Beyond healthcare settings, the COVID-19 pandemic accelerated the adoption of contactless smart hand-washing stations and sensor-based waste management systems [34]. These solutions reduce indirect contact with potentially contaminated surfaces and contribute to environmental sustainability by optimizing the use of water and detergents. Despite technological progress, relevant criticalities remain. Many healthcare workers perceive electronic monitoring systems as invasive or as punitive control tools, which fuels cultural resistance and privacy concerns [104, 106]. Furthermore, the costs of implementation and maintenance for IoT infrastructures can represent a significant barrier for smaller institutions [103, 34]. From a technical standpoint, many current systems still focus on the quantity of hygiene events without having a real capacity to evaluate the microbiological effectiveness of the performed action [103, 106].

The evolution of technologies for Human Activity Recognition (HAR) has accelerated significantly over the last decade. This progress is driven by the miniaturization of sensors and the increasing computational power of wearable devices. While HAR originally emerged for fitness and lifestyle monitoring, recent global health emergencies have shifted scientific attention toward critical public health applications. Scientific literature has explored various paths within the field of computer vision. For instance, Zhong et al. proposed a complex multi-camera system capable of recognizing seven specific hygiene actions [108]. Despite their accuracy, these video-based systems face significant obstacles such as high costs and limited coverage. Furthermore, they raise serious concerns regarding user privacy. In contrast, the use of inertial sensors (IMU) integrated into wearable devices offers a pervasive and privacy-respecting alternative. Early contributions in this domain relied on multiple high-sensitivity sensors typical of scientific instrumentation [109, 110, 111]. More recently, there has been a democratization of this technology. Lattanzi et al. demonstrated that it is possible to recognize unstructured hand washing with 95% accuracy using Deep Learning and 94% with standard techniques [40]. They achieved this by using only the accelerometer and gyroscope signals of a common commercial smartwatch. In this scenario, the smartwatch becomes a proactive actor capable of inducing virtuous behaviors

by monitoring the hygiene status of the user.

Expanding the view to the general field of HAR, the primary objective is to distill knowledge from raw data to identify behavioral patterns [112, 113]. Techniques are clearly distinguished based on the data source: video-based and sensor-based. While computer vision offers robustness in controlled environments [114, 115], the sensor-based approach is considered more flexible and economical for continuous monitoring. Modern wearable devices integrate microcontrollers (MCU) and inertial measurement units (IMU) sensors in small packages, enabling pervasive computing scenarios [64]. The data collected from smartphones [116] or smartwatches [117] is fed to classification algorithms capable of distinguishing between complex activities of daily living (ADL).

Traditionally, HAR relied on Machine Learning (ML) algorithms executed in the Cloud. This paradigm offers unlimited computational resources but requires continuous data transfer through the network [118, 119]. This results in high costs regarding latency and energy consumption for transmission. The current trend, called TinyML, aims to move inference directly onto edge devices [120, 121]. These tiny devices are characterized by strict constraints, such as having less than 1 MB of RAM and power consumption in the mW range. The advantages of this shift include high reactivity by eliminating network latency [122] and improved privacy since sensitive data remains on the local device [123]. Furthermore, the energy cost of local computation is often lower than that of radio transmission [64]. However, porting complex models to limited hardware is not trivial. While training usually occurs on centralized servers [124], inference requires aggressive optimizations. Strategies include selecting the most informative features to reduce input size [125], reducing the sampling frequency [126], and quantization [127].

Literature presents a dichotomy between the use of shallow models and Deep Learning models on embedded devices. In the past, authors implemented classical algorithms like Naive Bayes, Support Vector Machines (SVM), and Decision Trees on wearables [128, 129, 130]. These models are lightweight but require heavy manual feature engineering. The advent of Deep Learning allowed for direct learning from raw sensor data. Novac et al. compared supervised and unsupervised approaches on embedded systems [131]. Chen et al. (2021) provided a comprehensive overview of Deep Learning techniques applied to sensors [90]. Notable examples include the work of Alessandrini et al., who implemented a Recurrent Neural Network (RNN) on an embedded device to process heterogeneous data (PPG and accelerometer)[123]. Bhat et al. developed custom hardware capable of executing the entire HAR chain with a very low energy consumption of $22.4\mu J$ per operation [132].

In parallel, research has focused on the adaptability of complex architectures. Wang et al. proposed tools for designing efficient Multilayer Perceptrons (MLP) for microcontrollers [133]. Disabato et al. investigated the accuracy of CNNs on ARM Cortex-M7 processors [134]. Advanced model compression techniques have also been suggested for deploying DNNs on MCUs [135]. Of particular interest are works that integrate attention mechanisms. Wang et al. demonstrated that Recurrent Attention Networks (RAN) and attention-based CNNs outperform classical methods [136, 137]. Coelho et al. [138] and Mayer et al. [139] also confirmed the suitability of various deep learning models on low-power platforms. Despite progress, real-world energy consumption analysis remains

insufficient in many studies. Most research is limited to theoretical evaluations or simulations. However, innovative solutions based on adaptive architectures and hierarchical offloading are emerging. Samie et al. proposed a hierarchical classification process [140]. Similarly, Daghero et al. [141] use a two-stage approach where a decision tree handles easy classes and activates a more expensive 1D-CNN only for complex activities. Another approach is the Early Exit Neural Architecture Search (EExNAS), which allows a network to exit early if confidence is sufficient [142]. Rashid et al. applied this concept to eating activity recognition [143, 144].

A clear gap in the state of the art is the lack of systematic triggering mechanisms. Triggering refers to activating a system only when a specific event occurs. This approach is well established in signal processing, where it is used for data acquisition and synchronization [145]. Techniques range from simple threshold detection [146] to more advanced pattern recognition [147]. In wireless sensor networks (WSN), the use of wake-up triggers drastically reduces energy wasted during idle periods [148, 149]. In these systems, the main device is activated only when a meaningful signal is detected [150]. Applying these concepts to wearable-based HAR could represent a major step forward in improving device autonomy and energy efficiency.

Beyond hardware constraints, energy efficiency is deeply influenced by software design and code quality. Research indicates that inefficient programming practices, often named "energy code smells", can significantly degrade battery life through improper resource management, such as unnecessary object allocations or sensors remaining active longer than required [151, 152, 153]. While developers across different platforms show varying levels of awareness regarding these issues, the consensus in the literature is that native solutions generally outperform cross-platform or web-based abstractions (i.e., Progressive Web Apps) in terms of energy footprint [154, 155, 156].

In the context of wearable devices, these challenges are even more pronounced due to limited battery capacity and the demands of continuous sensing. Key strategies to mitigate high power consumption include: using lower-power alternatives for long-term monitoring and optimizing sensing intervals [157, 158], dynamically shifting heavy processing tasks (like complex activity recognition algorithms) from the wearable device to external edge or cloud services [159], and adopting frameworks specifically designed to detect and prevent abnormal sensor behaviors or runtime overhead [153, 160]. These software-level optimizations are crucial for ensuring that the monitoring system remains non-intrusive and operational throughout an entire work shift without requiring frequent recharging.

The most advanced challenge in HAR concerns the capability of models to adapt to individual users while relying on very limited labeled data. This issue, commonly described as data scarcity, is especially pronounced in fine-grained activities such as hand washing, where personal motor patterns, execution tempo, and subtle variations have a strong influence on the recorded signals. Subject-independent models, although robust and easy to scale, often struggle to account for these individual differences and instead smooth out user-specific traits that are semantically relevant. To overcome this limitation, Deep Metric Learning (DML) has gained increasing attention. Instead of performing direct classification, metric learning focuses on constructing a latent embedding space in which the distance between samples encodes their semantic similarity [161]. Such representations

allow the system to recognize new users or previously unseen execution styles from only a handful of labeled examples, thus enabling Few-Shot Learning scenarios. Within this framework, models such as Siamese Neural Networks (SNN) and Matching Networks have been extensively investigated, as they operate by explicitly comparing pairs or small sets of samples to estimate similarity and support personalized recognition with minimal supervision [162, 163]. These approaches are particularly suitable for wearable-based HAR applications, where acquiring large, user-specific annotated datasets is often impractical.

While DML addresses the problem of limited labeled data, it does not mitigate the temporal complexity of human activities. Many daily actions unfold over time and exhibit long-range dependencies that are difficult to model with traditional sequential architectures such as RNNs or LSTMs. This limitation has motivated the introduction of attention mechanisms in HAR. Attention-based models allow the network to selectively focus on the most informative parts of a sensor sequence, for example, the precise moments in which hands make contact or change motion patterns. Early works demonstrated that this selective focus leads to significant accuracy improvements compared to uniform sequential processing [137, 136]. Building on this idea, the adoption of Transformer architectures has marked a further step forward. By relying on self-attention rather than recurrence, Transformers can process entire time series in parallel and capture global temporal relationships more effectively. Although their computational and memory requirements still pose challenges for deployment on resource-constrained devices, ongoing research is increasingly oriented toward lightweight and efficient attention-based models that replace recurrence with pure attention to better model the semantics of human movement.

In parallel, Natural Language Processing (NLP) techniques have increasingly influenced human activity recognition, particularly through the adaptation of pre-trained transformer architectures based on Bidirectional Encoder Representations from Transformers (BERT). Drawing inspiration from language modeling, several methods reinterpret accelerometer or inertial measurements as sequences of discrete tokens, analogous to words within a sentence. Through self-supervised learning schemes, large models are trained on extensive collections of unlabeled sensor data by masking portions of the signal and learning to reconstruct them [131]. This training paradigm yields expressive and reusable embeddings that capture both fine-grained temporal dynamics and longer-range structural patterns of human motion. Once learned, these representations can be fine-tuned using only small labeled datasets for specific tasks, such as hand hygiene recognition, thereby substantially reducing the cost and effort associated with manual annotation.

Finally, the recent advancements of Large Language Models (LLMs) further extend this paradigm by enabling Zero-Shot Learning in multimodal settings. By associating textual descriptions of activities with sensory patterns, these models can recognize actions for which no explicit training examples were provided. A system can infer the meaning of a gesture from its semantic description, bridging the gap between language understanding and signal analysis. This convergence represents a promising direction toward more flexible, general, and user-adaptive HAR systems that can operate across contexts, users, and tasks with minimal supervision.

2.5 Autonomous Cleaning

The introduction of robotic systems for sanitization and maintenance represents an important technological evolution towards the objective of reducing biological risk, increasing operational resilience, and improving the measurable quality of cleaning. High-criticality healthcare environments, such as hospitals, assisted living facilities, and schools, have highlighted the structural limits of manual cleaning processes. These processes are characterized by high execution variability, poor traceability of operations, and a significant risk of exposure for operators [164, 165]. From a quality-oriented standpoint, robotics for sanitization extends well beyond the simple automation of established tasks. Instead, it represents a core element of integrated cyber-physical systems in which robotic platforms, IoT sensing infrastructures, and digital building models jointly operate to guarantee hygienic and sanitary standards that are both reproducible and objectively verifiable. In this context, the sector is evolving from predominantly manual practices toward autonomous or semi-autonomous solutions that combine advanced perception capabilities, intelligent navigation, and physical or chemical disinfection technologies, in line with the paradigms of predictive maintenance and adaptive management [166, 167].

Environmental perception and intelligent dirt detection represent the first level of technological integration. Modern cleaning robots must interpret their surroundings not merely as a traversable space, but as a set of functional surfaces with varying contamination levels. Unlike consumer-grade robots that rely on basic contact or acoustic sensors, advanced professional platforms employ computer vision systems to support data-driven operational decisions. Rather than requiring supervised training for every specific material, these systems can identify dirt by analyzing visual deviations against the regular texture of the floor. This allows for the autonomous construction of semantic dirt maps, often achieving detection rates near 90% [166]. Such high-fidelity environmental mapping is crucial for the transition toward dynamic cleaning models, as it enables the robot to prioritize interventions based on the actual hygienic state of the environment rather than a fixed schedule.

In high-risk contexts such as Intensive Care Units (ICU), technological evolution includes the use of autonomous sanitization robots equipped with biosensors capable of detecting pathogens, such as SARS-CoV-2, through rapid DNA/RNA amplification techniques. These systems can integrate disinfection technologies based on ozone and UV radiation, monitoring their levels to ensure the safety of both patients and operators [14, 168, 29].

Environmental perception capabilities must be accompanied by high precision in localization and navigation, especially in complex and dynamic environments such as hospitals, airports, or high-density offices. In this context, the adoption of LiDAR sensors and Simultaneous Localization and Mapping (SLAM) algorithms represents the technological reference standard [169, 167]. LiDAR enables the generation of two-dimensional or three-dimensional maps in real time with millimeter accuracy, which is an essential requirement for avoiding static and dynamic obstacles such as staff, patients, or mobile equipment [170].

A further advancement over simple navigation automation is represented by trajectory planning based on the disinfection dose. In this paradigm, the robot dynamically adapts its speed and path as a function of the accumulated UV-C dose on each surface, which

is calculated considering distance, incidence angle, and exposure time [171]. If a specific area has not received the dose necessary for the inactivation of the target pathogen, the control system, often implemented on Robot OS (ROS) platforms, modifies the trajectory to ensure adequate exposure [171, 167]. This approach is particularly effective in wall-follower strategies, which maximize the irradiation of vertical surfaces and the edges of obstacles that are typically more subject to contamination. The dosimetric map can also be integrated into the Digital Twin of the building as an information layer, allowing for the a posteriori validation of operations and the continuous optimization of sanitization cycles.

The effectiveness of sanitization robots depends largely on their integration into a connected digital ecosystem. The adoption of 5G technology, with latencies below 10 ms, enables real-time remote control and high-definition video monitoring even in teleoperation scenarios [164, 170]. Simultaneously, integration with IoT infrastructures allows for the continuous transmission of data regarding the status of the robot, disinfectant levels, battery charge, and air quality parameters, creating an information flow consistent with Building Management Systems (BMS) [169].

From the perspective of human-machine interaction, dedicated mobile applications allow operators to start sanitization cycles, monitor safety timers, and verify the status of operations from outside contaminated environments, thereby reducing the risk of exposure [165, 172]. In hospital contexts, the sanitization robot also tends to be configured as an intelligent logistical node, capable of performing transport functions for sterile materials, medications, or biological samples, thus contributing to the optimization of internal flows [173]. Finally, the presence of auto-docking systems and autonomous energy management allows for continuous and unattended operation. The robot is capable of monitoring its own energy status and independently returning to the charging station, making these solutions fully compatible with the predictive management models of smart buildings [169, 172].

2.6 Smart Monitoring for Restroom Hygiene and Maintenance

Regarding smart cleaning models and environmental quality management, attention naturally extends to high-intensity confined environments where sanitization assumes a critical role that is not only operational but also sanitary and perceptive. Among these, collective restrooms represent an emblematic case. The combination of high usage variability, the presence of biological and chemical contaminants, and the strong impact on the overall perception of building quality makes the adoption of static cleaning schemes insufficient [34, 19]. Scientific evidence shows that manual inspection and sanitization processes suffer from structural limitations similar to those observed in traditional cleaning models, including subjective bias, poor repeatability, difficulties in ensuring spatial and temporal uniformity of hygienic standards, and limited traceability of operations [35, 174]. In this context, a key driver for service evolution is the capability to formalize and quantify conditions that have traditionally been assessed qualitatively, such as perceived environmental

discomfort or odor presence, enabling their systematic integration into digital facility management platforms and their use as objective criteria for activating cleaning and ventilation actions.

Chemical air monitoring constitutes one of the pillars of this approach. In restrooms, the presence of compounds such as ammonia (NH_3), hydrogen sulfide (H_2S), and VOCs does not represent only a source of olfactory discomfort but is configured as an indirect indicator of hygienic status and environmental degradation. In this direction, the works of Zhou et al. introduce the concept of artificial olfaction applied to sanitization, demonstrating how electrochemical sensor arrays, combined with machine learning algorithms, are able to reliably distinguish between odors related to the physiological use of services and detergents or food residues [37, 175]. Similar results are reported by Yatabe et al., who show how chemosensitive resistors can help classify the comfort state of the environment with accuracies close to 98%, validating the hypothesis that olfactory comfort can be quantified in a robust and reproducible way [176].

Once the hygienic state of the environment is observable, the next step concerns the optimization of resources and the transition toward predictive maintenance strategies. Several studies demonstrate how the analysis of historical usage data allows for the anticipation of the degradation of hygienic conditions before it becomes perceptible to the end user. The Toilet Alarms system proposed by Turman-Bryant et al. represents an emblematic example. Through ensemble models, it is possible to predict the filling of waste bins and plan interventions only when actually necessary, obtaining significant reductions in operational costs and manual workload [39]. Predictive approaches based on Recurrent Neural Networks (RNN) further confirm the possibility of estimating the evolution of hygienic conditions with sufficient advance to enable preventive action [36].

Smart maintenance is not limited to the planning of interventions but also involves the overall resilience of the infrastructure. The self-diagnosing bathroom prototype developed by Cid et al. falls in this direction, as it uses chemical-physical sensors to continuously monitor the functioning of waste treatment systems and sends operational instructions directly to technicians through mobile applications [177, 178]. Such an approach reduces downtime and lowers the skill threshold required in the field. Parallelly, the physical automation of cleaning and sterilization operations represents a further level of service integration. Experimental studies show how even apparently marginal parameters, such as the duration of the automatic flush, can be scientifically optimized to maximize hygienic effectiveness while reducing resource consumption [179]. More advanced IoT systems allow for the dynamic activation of ventilation, deodorization, and sanitization cycles based on the detected gas concentrations, transforming the bathroom into a reactive and self-regulated environment [180, 181]. The integration of UV-C lamps, activated exclusively in the absence of occupants thanks to infrared sensors, further extends the concept of continuous sanitization, which reduces the bacterial load without compromising user safety [182].

An aspect that is often underestimated concerns the social dimension of cleaning. Literature highlights how the perception of hygiene varies significantly according to the cultural context and the type of user. Hybrid systems that combine objective sensor data and direct user feedback allow for the dynamic adaptation of operational thresholds, building a

tolerance spectrum specific to each environment [35, 183].

2.7 IoT in Environmental Monitoring

Indoor environmental monitoring is essential for quality-oriented cleaning, sanitization, and space management strategies. Continuous observation of indoor conditions helps ensure that hygienic actions are effective and applied at the right time. It also allows interventions to be aligned with real environmental risks. Monitoring systems make it possible to link cleaning activities to measurable health outcomes. This supports a shift from reactive and time-based management to more adaptive and evidence-based approaches. The need for systematic indoor monitoring is strongly related to the amount of time people spend inside buildings and the associated health effects. Indoor Air Quality (IAQ) is therefore a key factor for health, well-being, and cognitive performance, especially considering that people spend up to 90% of their time indoors [1].

Several studies show that indoor concentrations of chemical and biological pollutants can be much higher than outdoor levels. This is mainly due to limited air exchange and emissions from building materials, furniture, and cleaning products [28, 24, 3]. Carbon dioxide concentration is commonly used as an indicator of ventilation efficiency [25, 32]. Studies conducted in schools, offices, and healthcare facilities highlight a clear trade-off. Natural ventilation reduces CO_2 levels but allows outdoor particulate matter such as $PM_{2.5}$ and PM_{10} to enter. Air-conditioned environments, on the other hand, limit particulate matter but often show high CO_2 concentrations, sometimes up to 1680 ppm [6, 184]. Long-term exposure to pollutants like particulate matter, nitrogen dioxide (NO_2), and VOCs is especially harmful for vulnerable groups, including children [184, 2]. VOCs and aldehydes released by building materials and aggressive cleaning agents are among the main causes of SBS. This shows that poorly designed cleaning practices can worsen indoor air quality by releasing irritating substances [2, 185, 5].

In this context, IoT technologies allow continuous and distributed monitoring of key environmental parameters. These include $PM_{2.5}$, PM_{10}, CO_2, temperature, relative humidity, and VOCs [186]. Most IoT-based IAQ systems focus on direct measurements and on checking compliance with regulatory limits [187]. Recent studies propose more advanced approaches based on virtual sensing, in which data fusion and analytical models are employed to estimate complex biological variables, such as airborne bacterial load, from standard physical measurements [188]. Such approaches support a more proactive and risk-based management of indoor environments, enabling cleaning and sanitization actions to be planned according to real exposure conditions rather than fixed schedules.

Despite the benefits offered by IoT systems for environmental monitoring, their sustainable deployment is significantly limited by energy constraints. Energy consumption represents a primary challenge for the IoT ecosystem and demands solutions specifically designed for efficiency [189]. Experience in the field of Wireless Sensor Networks (WSN) has demonstrated that data transmission is the most energy-intensive operation, often surpassing both data acquisition and local processing [148]. This reality has prompted the scientific community to develop strategies aimed at reducing the volume and frequency of communications.

To address this problem, several data traffic reduction strategies have been proposed, including compression, quantization, and aggregation techniques [190]. Prediction-Based Data Collection (PBDC) and Model-Based Sensing have emerged as particularly effective solutions for periodic monitoring scenarios [149]. In these approaches, a time series prediction model is shared between the IoT node and the collection server. When the model accurately describes the data trends, the number of transmitted packets and the resulting energy consumption can be reduced by up to 99% [191, 192, 193, 194].

Over the last two decades, numerous models have been explored to support PBDC and model-driven data acquisition strategies [195]. Statistical and probabilistic models, including those based on multivariate Gaussian distributions [196] and generic probabilistic models [197, 198], offer high accuracy but require experts to manually define data characteristics. Simpler approaches like constant models [199] are computationally lightweight, although they are only applicable in stationary contexts. Other studies have proposed linear and autoregressive models to improve efficiency. Tulone and Madden demonstrated the effectiveness of autoregressive models in reducing communications [200, 201]. While linear regression [149] and Derivative-Based Prediction (DBP) [202] represent simple solutions, they are sensitive to noise and cannot capture strongly non-linear dynamics. Kalman filters [203, 204] have been widely adopted for their computational efficiency and ability to operate in noisy systems, yet they show limits in modeling complex non-linear correlations.

Adaptive sampling represents a complementary line of research where both transmission and sampling frequency are dynamically adapted. Gedik et al. introduced an approach based on the dynamic selection of sampler and non-sampler nodes [205]. Recent studies propose algorithms based on finite state machines to maximize the sampling rate without compromising battery life [206], or methods that optimize the acquisition interval based on the distance between consecutive samples [207]. Other works suggest skipping sampling entirely when the estimated information loss is minimal [208], a strategy successfully applied in wearable systems for physiological monitoring [209].

With the exponential growth of data generated by IoT systems, classic forecasting methods based on Support Vector Machines (SVM), Random Forests (RF), and Backpropagation (BP) have shown limits in managing high-dimensional and non-linear multivariate time series [210, 211]. To overcome these limits, deep learning models such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU) have been introduced. These models have achieved significant results in predicting complex time series [212, 213]. In the IoT domain, these techniques have been primarily used for reconstructing missing or corrupted data, which improves the overall reliability of monitoring systems [214, 215, 216, 217].

Due to their high computational cost, these models were historically confined to the cloud or fog levels of the IoT architecture. To optimize resource use, offloading mechanisms based on deep reinforcement learning have been proposed to dynamically decide where to execute specific computational tasks [218, 219]. However, executing deep learning models directly on the edge is progressively becoming a concrete possibility. Lalouani et al. demonstrated the use of LSTM networks on wearable ECG sensors to predict and skip redundant samples [209]. Similarly, Capellacci et al. showed how low-cost IAQ sensors can integrate deep learning-based PBDC algorithms to reduce transmissions by up to 96%

without compromising monitoring quality [220].

Existing literature clearly demonstrates the effectiveness of PBDC, adaptive sampling, and artificial intelligence in reducing energy consumption. However, many studies are limited to evaluations in simulated environments or theoretical energy estimates. Some works have explored the extension of sleep mode periods through the use of prediction intervals [221, 222]. Other research combines compression techniques like Principal Component Analysis (PCA) with autoregressive models [223, 224]. Despite these developments, the number of studies that validate such approaches on real and highly constrained hardware remains limited.

2.8 Summary

In summary, the analysis of the scientific literature highlights a decisive trajectory toward the digitization of cleaning and facility management. The core building blocks, IoT sensing, advanced machine learning, and edge computing, are well-established, enabling a move away from static cleaning schedules toward dynamic, context-aware interventions. However, significant challenges remain. Existing research treats environmental monitoring, vision, and HAR as separate topics, leaving the potential of fusing these heterogeneous data streams largely unexplored. This lack of a unified vision prevents the industry from achieving a truly transparent and verifiable hygiene standard, as the interaction between occupant behavior and microbial load is still mostly inferred rather than measured.

This thesis addresses these limitations by proposing a framework that bridges the gap between manual protocols and digital traceability through three original contributions. First, the research explores how to estimate airborne bacterial loads from environmental proxies using deep learning and low-cost IoT nodes, effectively replacing expensive and slow manual sampling. This is complemented by the use of computer vision to extract real-time occupancy and flow metrics directly on the device, ensuring data privacy while optimizing cleaning frequency. Finally, the framework introduces the use of wearable sensors to track cleaning activities and bridge the manual traceability gap, allowing for the validation of hygiene protocols and the quantification of the contamination generated by occupant movement. All proposed models are designed to operate within the strict energy constraints of hardware, ensuring that the system is both analytically accurate and operationally sustainable for continuous monitoring.

Chapter 3

IoT for Indoor Monitoring

In modern days, ensuring high indoor environmental quality is no longer just a matter of comfort but a fundamental requirement for public health and disease prevention. Since people spend the majority of their time in confined spaces, these environments can become primary vectors for respiratory infections and biological cross-contamination. However, current scientific literature and commercial solutions primarily focus on monitoring basic physico-chemical parameters as isolated metrics, often failing to establish a direct, actionable link to public health risks and microbial safety. Internet of Things (IoT) technologies represent the essential bridge to overcome this limitation.

This chapter demonstrates how advanced IoT applications can bridge the gap between simple environmental sensing and comprehensive biological safety. The discussion is organized around two primary research contributions. The first is the development of a virtual sensor that transforms standard, low-cost IoT nodes into intelligent sentinels. By employing machine learning models, this system estimates airborne bacterial loads, providing a continuous monitoring capability that overcomes the prohibitive costs and time lags associated with traditional manual sampling. The second contribution is a predictive technique designed to optimize the trade-off between data resolution and energy consumption. By intelligently modulating the sampling frequency, these methods extend the lifespan of the sensing infrastructure without compromising the quality of the monitoring service, ensuring the sustainability of long-term biological risk assessment.

3.1 Background

The growing adoption of Internet of Things (IoT) technologies has brought significant transformations across multiple domains, including environmental monitoring, smart building management, agriculture, and transportation. In these contexts, IoT devices leverage embedded sensors to gather real-time data, allowing for continuous and detailed observation of environmental conditions. This persistent monitoring enables more efficient operations, promotes sustainability, and enhances the comfort and well-being of occupants.

The origins of the IoT concept date back to the early 1980s, predating the introduction of the term itself. One of the first known connected devices was a modified Coca-Cola

vending machine developed at Carnegie Mellon University. The goal was to monitor inventory levels and internal temperature remotely. This project already showed a key benefit of IoT systems, namely improved operational efficiency [225]. During the 1990s, other experimental systems further explored this idea. A notable example is John Romkey's "smart" toaster, which demonstrated that home appliances could be connected to the internet using TCP/IP protocols. The term "Internet of Things" was formally introduced in 1999 by Kevin Ashton. He described a vision in which computers could observe the physical world directly through sensors, without constant human input [226]. Early systems were often classified as Machine-to-Machine (M2M) solutions. These relied on point-to-point connections and proprietary networks designed for specific tasks. Modern IoT systems, in contrast, are based on network-oriented and cloud-native architectures. This approach allows for scalability and integration with advanced data analytics and learning models [227].

To evaluate the effectiveness of modern IoT monitoring solutions, it is necessary to examine their core functional components. A typical IoT ecosystem can be described through five main layers [228]:

1. Sensors and Actuators form the "things" layer. They capture environmental data, such as gas concentrations or occupancy counts, and enable the execution of physical actions.

2. Processors perform local computation, spanning from low-power microcontrollers to more capable embedded platforms that handle data processing and basic security tasks.

3. Gateways oversee the transfer of data between devices and servers. This communication may use wired or wireless links, relying on protocols such as MQTT or ZigBee.

4. Applications are either cloud-based or local platforms that analyze data and present actionable insights to users, for instance, through dashboards for facility management or home automation.

5. Data Storage refers to databases used to store historical data. These are essential for long-term analysis and for training machine learning models.

In addition to these functional layers, data processing in IoT systems is usually distributed across three hierarchical levels: (i) Edge, (ii) Fog, and (iii) Cloud, as shown in Figure 3.1. The Edge layer includes sensors, devices, and microcontrollers placed directly at the data source. In traditional designs, these elements mainly transmitted raw data. More recent approaches adopt edge computing techniques where data is partially processed locally to reduce latency and limit network traffic. The Fog layer acts as an intermediate level between the Edge and the Cloud. Fog nodes include gateways, routers, or local servers offering computing, storage, and networking resources close to end devices. Their main role is to filter and aggregate data, sending only relevant information to the cloud. This reduces jitter and improves responsiveness in real-time applications. The Cloud layer consists of centralized servers and data centers. It provides large-scale storage and high

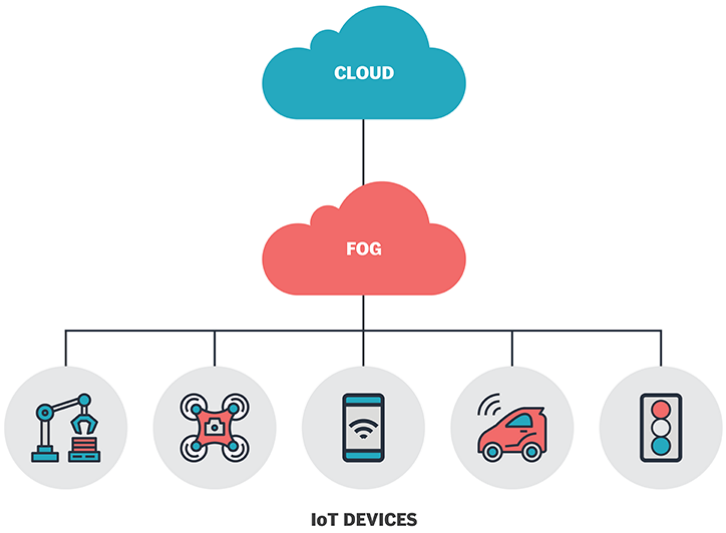


Figure 3.1: Schematic representation of the three layers of IoT: Edge, Fog, and Cloud. Credit: Kanda Euatham.

computational power. This makes it suitable for long-term analytics and training complex models. However, exclusive reliance on the cloud can increase latency and transmission-related energy consumption.

3.1.1 Indoor Environment Monitoring

The application of IoT technologies to environmental monitoring has become a focal point of research. Given that humans spend a significant portion of their time indoors, the continuous tracking of parameters such as temperature, humidity, and pollutant concentrations is imperative for ensuring occupant health and comfort. IoT systems enable real-time data collection from distributed sensor nodes, providing opportunities to optimize environmental conditions while simultaneously minimizing energy consumption [229].

The literature reports extensive work on the development of tools for monitoring environmental parameters. Systematic reviews of IoT-based indoor air quality (IAQ) systems indicate that many studies rely on low-cost prototyping platforms, such as Arduino or Raspberry Pi, to collect environmental data [230]. The applications of these systems range from general environmental monitoring to highly specific indoor use cases. For instance, Bianconi et al. [231] implemented an edge-based IoT system in Savona, Italy, aimed at monitoring citizen well-being over extended periods and responding to critical events. In a similar vein, Narayana et al. [232] proposed a real-time platform that gathers data across air, water, and energy domains, supporting broader sustainability objectives. In indoor environments specifically, systems have been developed to monitor multiple pollutants simultaneously, including $PM_{2.5}$, CO_2, and VOCs [186]. Some of these solutions incorporate predictive functionalities to generate alerts under hazardous conditions [233], while others, such as Sung et al. [234], employ standardized air quality indices to simulate how

environmental factors affect indoor air quality.

With the rise of AIoT, data-driven approaches are gaining momentum. Models such as LSTM networks are being applied to predict future CO_2 concentrations, enabling pre-emptive ventilation actions. Furthermore, multi-headed Convolutional Neural Networks (CNNs) have been employed to enhance prediction accuracy through transfer learning, boosting system efficiency even in data-scarce environments [235].

Despite these advancements, traditional IoT-based monitoring systems face significant hurdles. A primary concern is energy consumption. Systems often rely on continuous data collection and transmission, which depletes batteries rapidly in resource-constrained edge devices [236]. Additionally, low-cost sensors are prone to drift, raising issues regarding continuous calibration, standardization, and data reliability. To develop robust, energy-efficient algorithms, high-quality datasets are foundational. The literature reveals a rapid expansion in the availability of IAQ datasets capturing parameters like PM_{10}, CO_2, and noise levels [220, 237, 238, 239].

3.1.2 IoT Energy Constraints and Reliability

The IoT is fundamentally reshaping modern society by embedding intelligence into everyday objects. This transformation affects a wide range of sectors, including healthcare, agriculture, industry, and urban infrastructure [240, 241]. However, the rapid growth of IoT deployments comes with a high environmental cost. As of 2024, more than 16 billion IoT devices are estimated to be in operation worldwide, with projections exceeding 30 billion by 2030 [242]. This large expansion brings a substantial energy footprint. Recent studies suggest that IoT systems already consume hundreds of terawatt-hours (TWh) of energy each year [243]. If current trends continue, IoT technologies could account for as much as 25% of global energy demand by 2030 [244]. For this reason, reducing energy consumption at both hardware and software levels is essential to ensure the long-term sustainability of the IoT ecosystem.

Although sensing and local processing require energy, data transmission represents the dominant contributor to power consumption in most IoT applications, particularly when wireless communication is used [148, 245]. Continuous data collection places a heavy burden on edge devices, which is especially critical in resource-constrained environments such as battery-powered sensor nodes [246, 247]. To address this issue, several energy-saving strategies have been proposed in the literature. Duty Cycling techniques periodically switch communication modules into low-power sleep states to conserve energy. Adaptive Sampling approaches dynamically adjust the sampling rate based on changes in the monitored environment. For example, Giordano et al. proposed an energy-aware algorithm based on a Finite State Machine (FSM) to maximize sampling while preventing battery depletion [206]. Similarly, Ben-Aboud et al. introduced lightweight algorithms to optimize sampling intervals in low-power devices [207]. Other research efforts focus on Task Offloading, where computational workloads are transferred from constrained devices to fog or cloud infrastructures to reduce local energy consumption [218, 219]. Despite their effectiveness, these approaches often present notable limitations. Many rely on static configurations and lack adaptability to dynamic environmental conditions. Others reduce energy consumption at

the expense of data accuracy or fail to optimize sensing and communication costs. As a result, there is a strong need for intelligent and adaptive frameworks capable of significantly reducing energy usage without compromising system performance or data fidelity.

To address the trade-off between energy efficiency and data accuracy, Prediction-Based Data Collection (PBDC) has emerged as an effective solution. PBDC aims to reduce the number of transmitted samples while preserving the original sampling frequency required by the application [202]. This approach, also known as Model-Based Sensing [149], relies on a dual-prediction mechanism. The same forecasting model is deployed both on the edge device and on the central server (cloud). The operational workflow of PBDC is simple and efficient. When a new sensor sample is collected, the edge device compares the measured value with the value predicted by its local model. If the difference remains within a predefined error tolerance, the device suppresses the transmission. In this case, the server detects the absence of data and reconstructs the sample using its own copy of the prediction model. If the measured value deviates significantly from the predicted one, the sample is transmitted to the server to preserve data accuracy and update the model. By transmitting data only when predictions fail, PBDC significantly reduces bandwidth usage and energy consumption. This strategy preserves the high temporal resolution of the dataset while avoiding unnecessary transmissions [248]. When the prediction model accurately captures the underlying data trends, network traffic and associated energy costs can be reduced by up to 99% [191, 192]. For these reasons, PBDC represents a key technique for sustainable and energy-efficient IoT deployments.

3.1.3 Integration of IoT and Artificial Intelligence

The integration of IoT systems with Artificial Intelligence (AI) has introduced a major paradigm shift in how data is collected, processed, and exploited across modern infrastructures. This convergence, commonly referred to as the Artificial Intelligence of Things (AIoT), marks the transition from networks of passive connected devices to distributed systems capable of autonomous decisions and predictive analysis [249]. In this context, IoT infrastructures increasingly resemble a decentralized digital "nervous system" rather than simple sensing networks.

Since the early days of IoT research, several mathematical and statistical models have been explored to interpret sensor data. These early approaches were designed with computational simplicity in mind, making them suitable for very limited hardware platforms. However, this efficiency was often achieved at the expense of predictive flexibility and modeling accuracy. Initial solutions mainly relied on probabilistic models [197, 198]. These methods approximate sensor readings within a predefined confidence interval and require limited computational resources. At the same time, they often depend on domain knowledge, as data characteristics must be manually encoded by experts. Even simpler approaches, such as constant models [199], assume that sensor values remain stable over time. While extremely lightweight, these models are applicable only in strictly stationary conditions and perform poorly in dynamic environments.

To better handle temporal variations, linear regression models [200, 149] and autoregressive (AR) models [201] were introduced. These techniques provide improved modeling

of time-dependent data compared to constant or probabilistic approaches. However, linear models are unable to capture the nonlinear relationships that are common in environmental sensing data. Autoregressive models, on the other hand, are sensitive to noise, prone to overfitting, and typically assume periodic or highly regular signals. Kalman Filters represent one of the most advanced traditional techniques for sensor data estimation [204, 203]. They are particularly effective in tracking dynamic systems under noisy conditions and can be executed efficiently on resource-constrained devices. Despite these advantages, standard Kalman Filters show reduced accuracy when dealing with nonlinear correlations and depend strongly on accurate initialization of the system state. From an energy perspective, these models are highly efficient to execute but often lack the accuracy required to be used in the PBDC framework.

With the introduction of modern AI tools, research focus gradually shifted toward Machine Learning (ML). Classical ML techniques, including Support Vector Machines (SVM), Random Forests (RF), and backpropagation-based neural networks, began to replace purely statistical models for time-series forecasting [210, 211]. Compared to traditional approaches, these models provide a clear performance improvement, as they are able to capture nonlinear relationships and more complex data patterns. ML methods are particularly effective in analyzing sensor data, enabling more accurate anomaly detection and improved large-scale data analysis. However, the rapid expansion of IoT deployments has resulted in the continuous generation of massive data volumes. With billions of devices producing data streams in real time, even traditional ML approaches become insufficient [250]. As IoT systems scale to this level, the manual feature engineering required by classical ML models becomes a major bottleneck. In addition, classical ML techniques often struggle to address the so-called "6Vs" of IoT Big Data: Volume, Velocity, Variety, Veracity, Variability, and Value [249]. These limitations have motivated the adoption of more advanced learning paradigms capable of handling large-scale, heterogeneous, and high-velocity data streams.

Recent advances in Deep Learning (DL) provide effective solutions to the limitations of classical machine learning in large-scale IoT systems. DL models have shown strong capabilities in learning complex temporal dynamics directly from raw sensor data. In particular, Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), as well as Temporal Convolutional Networks (TCNs), have demonstrated high accuracy in modeling time-dependent patterns and forecasting sensor data streams [251, 252, 253]. Deep Neural Networks (DNNs) differ from shallow learning models mainly due to their ability to perform automatic feature extraction. This property significantly reduces the need for manual feature engineering and enables robust learning even in noisy and complex sensing environments [249]. As a result, DL models are particularly well-suited for large-scale and heterogeneous IoT data. In environmental monitoring applications, RNN-based architectures are among the most widely adopted solutions. They are commonly used for tasks such as predicting when indoor environments exceed hygiene thresholds based on occupancy and gas concentrations, or forecasting air quality trends over time [38].

A major limitation of many IoT systems is their strong dependence on centralized cloud infrastructures. This architectural choice introduces non-deterministic latency and

high bandwidth usage, which are unacceptable for mission-critical or real-time applications [254]. The integration of artificial intelligence directly at the edge, referred to as Edge AI, offers several important advantages. Local inference enables very low response times, which is crucial for real-time feedback and safety-related operations [250]. Processing data on-device also reduces network traffic, as only relevant events or alerts are transmitted to the cloud. Moreover, sensitive information can be analyzed locally, improving privacy and reducing exposure risks during data transmission [255].

Deploying deep learning models on microcontrollers (MCUs) presents substantial challenges due to the severe hardware limitations. Many MCUs feature less than 512 KB of on-chip RAM, which is typically insufficient to run standard neural networks efficiently [256]. To overcome these constraints, hardware-aware optimization techniques are commonly applied. These include network compression through pruning, reducing weight precision from 32-bit floating-point to 8-bit integers, and adopting approximate computing methods [256]. Such strategies enable advanced inference directly on constrained hardware, supporting autonomous intelligent agents capable of functioning reliably even in the absence of continuous internet connectivity.

3.2 Research Contribution

The transition from theoretical monitoring frameworks to operational systems requires a rigorous methodology that balances data accuracy with hardware efficiency. To move beyond static hygiene protocols toward dynamic, health-centered management, it is necessary to integrate intelligent sensing nodes that can process complex data at the edge. The following sections detail the technical architecture designed to bridge the gap between simple environmental sensing and real-time biological risk assessment.

3.2.1 IoT Virtual Sensor for Indoor Monitoring

The effectiveness of cleaning and sanitization protocols depends heavily on the ability to observe environmental conditions as they evolve. Conventional, schedule-based hygiene strategies are often disconnected from the actual biological state of indoor spaces, leading either to periods of insufficient protection or to unnecessary and inefficient use of resources. The adoption of a dedicated IoT-based monitoring platform helps to overcome these limitations by enabling continuous observation of key ambient parameters and by supporting the indirect estimation of microbial loads. This capability lays the groundwork for truly data-driven sanitization strategies, allowing hygiene management to shift from a purely reactive approach to one focused on timely prevention and risk mitigation.

The sensing node architecture is designed to balance affordability with advanced computational functionality. A key component is a virtual sensor that uses deep learning to capture non-linear relationships between environmental variables and microbial concentrations. This approach converts raw environmental measurements into actionable biological insights without requiring costly, specialized instruments.

The system encompasses both hardware and software layers. This includes the selection of environmental sensors, the development of custom firmware for efficient data acquisition,

and a machine learning pipeline that ensures the virtual sensor produces accurate and robust estimates across a range of indoor conditions.

The hardware architecture of the proposed IoT node is depicted in Figure 3.2, while Figure 3.3 shows a picture of the actual device installed in one classroom. The device has been designed as a multi-parametric environmental monitor integrating a comprehensive suite of low-cost sensors capable of capturing a wide range of physical and chemical parameters. This deliberate hardware selection ensures that the overall production of the virtual sensor remains affordable, guaranteeing both economic viability and high scalability for widespread deployment.

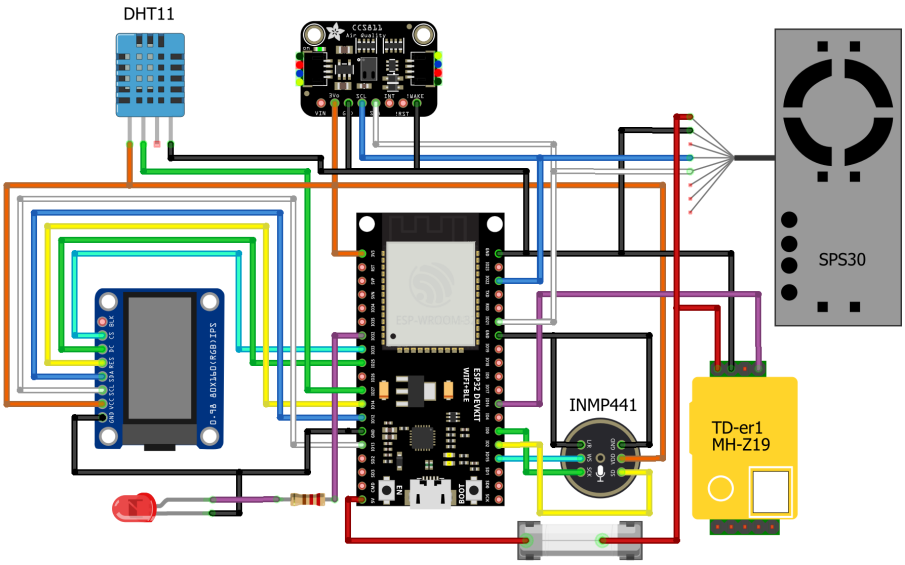


Figure 3.2: Schematic of the IoT device together with the environmental sensors connected to the main development board.

The sensing subsystem is composed of five dedicated modules, each focused on monitoring a specific environmental parameter. Carbon dioxide levels are measured using a Non-Dispersive Infrared (NDIR) sensor. This optical technique determines CO_2 concentration by measuring the absorption of infrared light by gas molecules, which correlates directly with their density. The sensor operates across a range of 0–5000 ppm and communicates with the microcontroller via UART and PWM interfaces, providing reliable digital data transfer.

Airborne particulate matter is assessed using a laser-scattering sensor. The device projects a laser beam to illuminate suspended particles, while a photodetector measures the scattered light to estimate particle concentration. It accurately detects particles from ultra-fine $PM_{1.0}$ to coarse PM_{10}, even at low concentrations, and transmits data through standard UART or I^2C interfaces.

Overall air quality is complemented by a multi-pixel metal-oxide (MOx) gas sensor capable of detecting a broad range of gaseous compounds. By monitoring Total Volatile Organic Compounds (TVOC) and estimating equivalent CO_2 (eCO_2) levels, the sensor

Figure 3.3: Photograph of the actual IoT sensor deployed in a classroom.

provides a comprehensive indoor air quality index, useful for identifying pollutants from paints, cleaning agents, and furniture.

Ambient temperature and relative humidity are captured using a combined digital sensor. Operating at temperatures between $0°C$ and $50°C$ and a humidity range of 20–90% RH, with a sampling frequency of 1 Hz. A single-wire communication protocol simplifies the circuit design and reduces the number of GPIO required.

Acoustic noise is monitored through a high-sensitivity digital microphone, covering the full human hearing range from 20 Hz to 20 kHz. Optimized for low power consumption, it transmits either raw audio or computed noise levels via the I^2C bus, enabling continuous assessment of environmental noise.

The platform is designed with usability and extensibility in mind. Multiple communication buses are exposed from the MCU, allowing additional sensors to be integrated without hardware modifications. For real-time feedback, the board includes two visual interfaces: a red LED that signals system states and errors (i.e., Wi-Fi failure or sensor timeout) using programmable flashing patterns, and a 0.96-inch display that cycles through current air quality readings, providing occupants with instant environmental information.

Software stack

The firmware for the IoT sensor node is designed as a reliable and fully configurable platform, with a strong focus on low power consumption, fast response times, and future expandability. These requirements are essential to support continuous sensing, real-time processing, and long-term integration of additional functionalities without major architectural changes. The software stack is developed using the Espressif IoT Development Framework (ESP-IDF) [257], which provides direct access to the hardware abstraction layer and peripheral drivers of the microcontroller. The system is built on top of the

FreeRTOS real-time operating system [258], enabling deterministic task execution and efficient resource management.

Unlike general-purpose operating systems for personal computers, a Real-Time Operating System (RTOS) is designed for embedded environments where precise timing and efficient resource management are essential. RTOSs are commonly used in industrial automation, robotics, telecommunications, and any application requiring predictable response times. FreeRTOS, in particular, operates as a lightweight kernel that is statically linked into the firmware, providing only the core services needed for task scheduling and synchronization while avoiding the overhead of a full-featured OS. Its strength lies in supporting multitasking on microcontrollers, enabling complex behaviors to be divided into smaller, independent tasks. In this implementation, the system adopts a cooperative multitasking model. Unlike fully preemptive scheduling, where the kernel interrupts tasks according to time slices, each task voluntarily yields control when idle or waiting for I/O. This approach minimizes context-switching overhead and gives fine-grained control over task execution order and timing, which is especially beneficial in resource-constrained embedded systems.

The firmware is written entirely in C and C++ to maximize execution efficiency and maintain direct control over low-level hardware resources. The software architecture exploits the dual-core capabilities of the microcontroller by coordinating three primary tasks that run concurrently under the FreeRTOS scheduler. The core component is the Measurement task, which implements the main application logic, including environmental data acquisition, signal processing, and data transmission. In parallel, the Telnet task provides a lightweight interface that enables local configuration and real-time debugging during development and deployment. Finally, the OTA task runs as a background service, continuously monitoring for firmware updates, thus ensuring device maintainability and upgradability without requiring physical access. Figure 3.4 presents the high-level execution flow of the firmware and illustrates the interactions among these concurrent tasks.

The Telnet task is responsible for system management and provides a remote Command-Line Interface (CLI) accessible via the network. By listening for incoming TCP connections on port 23, this service allows administrators to interact with the device and execute commands similarly to a conventional desktop terminal, thus removing the need for direct physical access to the hardware. The internal command parser is based upon a static registry of supported operations. As illustrated in Listing 3.1, each valid command is defined by a `telnet_command` C structure, which contains three elements: a unique string identifier for the command (`name`), a callback function (`run`) implementing the command logic, and a human-readable description (`help`) used to generate contextual help messages for the user.

During firmware initialization, a static list of all available commands is allocated. At runtime, user input received through the Telnet interface is parsed and compared against the `name` field of each registered command. If a match is found, the corresponding `run` callback is executed; otherwise, the system returns a *"command not found"* message. This interface allows dynamic adjustment of critical system parameters, such as sensor calibration ranges, sampling intervals, and the device identifier. To ensure settings persist across power cycles and reboots, all modifications made through the CLI are saved to the device's Non-Volatile Storage (NVS) and automatically reloaded during startup.

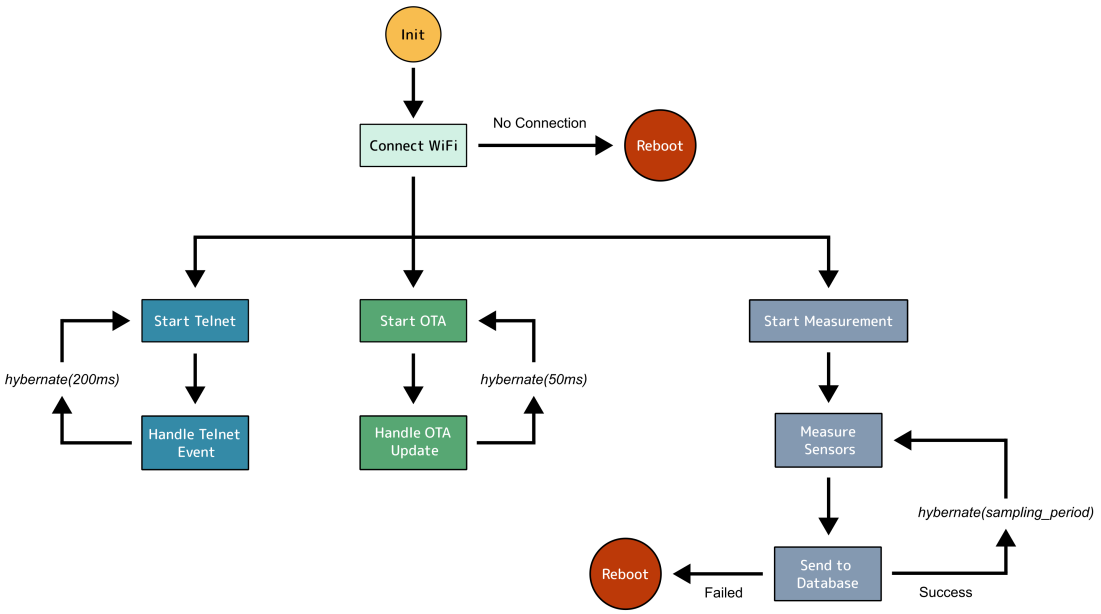


Figure 3.4: The main execution flows of the software stack of the IoT sensor.

```
typedef void (*cmd_on_run)(char *cmd);

struct telnet_command {
  char       *name;
  cmd_on_run run;
  char       *help;
};
```

Listing 3.1: C structure representing a generic telnet command.

To allow device maintenance without direct physical access, the firmware integrates a dedicated Over-The-Air (OTA) update mechanism. This functionality enables remote deployment of new firmware images via a Wi-Fi connection, removing the need for manual access to the microcontroller for reflashing operations. The OTA service operates as a background task that listens for incoming update requests on port 3232. Upon detecting a valid connection, the system initiates the update sequence by downloading the firmware binary and verifying its integrity and authenticity before installation. To ensure update reliability, the flash memory is organized using an A/B partitioning strategy: one partition hosts the currently running firmware, while the second is reserved for the incoming update. After the update completes successfully, the boot configuration is updated to select the new partition, followed by a controlled system reboot. This design ensures continuity of operation during firmware upgrades and provides a robust fallback mechanism, allowing the device to recover safely in case of update failure or image corruption.

The Measurement task serves as the central operational component of the firmware,

managing the full sensing pipeline while implementing energy-efficient logic. Execution follows a sequential workflow designed to minimize the active duty cycle of both the microcontroller and its peripheral sensors.

Upon waking from a low-power state, the task powers the required peripherals and collects raw measurements from all environmental sensors. Sensor-specific calibration factors are applied to convert these readings into accurate real-world values. Once acquired, the measurements are immediately transmitted to the remote server. After this process, the microcontroller and all peripherals enter hibernation mode, effectively disabling the CPU and primary clocks until the real-time clock (RTC) triggers the next wake-up event. This power-saving approach introduces a trade-off: during hibernation, OTA and Telnet services are inactive, temporarily preventing maintenance or firmware updates. Consequently, updates must be scheduled during active periods, or sleep intervals adjusted to allow access.

System reliability is reinforced through multiple watchdog checkpoints, denoted as *Reboot* states in the flow diagram. These checkpoints serve as fail-safe mechanisms, automatically restoring operation in case of unrecoverable faults. During boot, the firmware performs hardware initialization and attempts to connect to the configured Wi-Fi network. If the connection fails, a delayed software reset is scheduled to avoid rapid reboot loops during extended outages. Once a stable network connection is established, the firmware initializes the three main tasks (Telnet, OTA, and Measurement), entering normal operation. A similar watchdog mechanism operates during the measurement loop itself. Repeated failures in transmitting sensor data trigger a forced reboot, resetting the network stack and application logic, enabling recovery from transient communication or software faults.

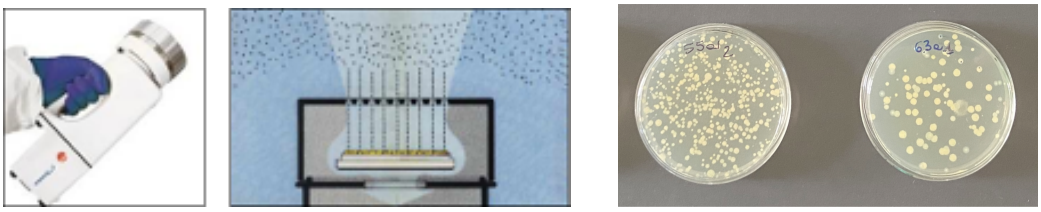
Estimation of the Aerial Indoor Biological Contamination

After defining the hardware and software architecture of the IoT node, the focus shifts to the implementation of the intelligent layer required for estimating aerial bacterial load. This transition represents the functional evolution of the device from a simple data logger into a virtual sensor, where, from raw environmental metrics, the bacterial load is inferred. To operationalize this transition, it is essential to define the specific dynamics of indoor biological contamination that the system is designed to monitor.

The presence of microorganisms in indoor environments represents a significant public health concern, closely associated with respiratory diseases and Sick Building Syndrome. Occupant exposure occurs primarily through two pathways: inhalation of airborne biological particles and contact with contaminated surfaces. Airborne microorganisms are carried within droplets of varying sizes. Fine droplets can remain suspended in the air for extended periods, increasing the likelihood of being inhaled. In contrast, larger droplets settle more rapidly under gravity, accumulating on floors, furniture, and other surfaces. When microorganisms adhere to dust particles, the potential for indirect transmission rises, as these particles act as vectors for contamination.

The microbial contamination of indoor air is commonly quantified as Colony Forming Units per cubic meter (CFU/m^3). This quantity is measured using two standard sampling techniques, illustrated in Figure 3.5: (i) active sampling, and (ii) passive sampling.

Active air sampling involves drawing a known volume of air at a controlled flow rate and



(a) (b)

Figure 3.5: Traditional methods for sampling airborne bacterial load: active sampling (a), and passive sampling (b).

depositing it directly onto a solid culture medium using devices such as Surface Air System (SAS) samplers. This approach enables precise and reproducible measurements of airborne microorganisms. In contrast, passive sampling uses settle plates containing a culture medium, which are exposed to the environment for a fixed period, allowing particles to settle naturally under gravity. While easier to implement, passive sampling produces less controlled and more environment-dependent results.

Following collection, culture plates are typically incubated for three to four days to allow microbial colonies to grow to a visible size for manual counting. Although microbiologically accurate, both methods are discontinuous and involve significant delays before results are available, highlighting the need for alternative approaches that can provide continuous, near-real-time indicators of biological contamination.

Since the goal of this study is to estimate a continuous target variable, namely the bacterial concentration expressed as CFU/m^3, the problem is formulated as a regression task. The estimation is performed using a set of independent environmental measurements that can be easily collected through low-cost sensors. The selected variables include carbon dioxide (CO_2), temperature, relative humidity, particulate matter ($PM_{2.5}$, and PM_{10}), and total volatile organic compounds (TVOC). These quantities are not direct measures of microbial concentration. However, they are either correlated with bacterial presence or influenced by the same environmental and anthropogenic factors that affect microbial growth and dispersion. The CO_2 is mainly produced by human respiration. For this reason, it is a good indicator of human occupancy and activity. Since humans are also a primary source of bioaerosols, this value shares a common production source with airborne bacteria, even though its diffusion mechanisms are different. TVOCs are also largely associated with human presence and activities, such as breathing, movement, and the use of cleaning products. Similar to CO_2, their emissions originate from the same sources that contribute to microbial release, although they spread differently in the environment. Temperature and humidity are well-known factors influencing bacterial survival and growth. Variations in their values can directly affect microbial metabolism and reproduction rates, making it a relevant variable for bacterial load estimation. Particulate matter plays a specific role as a potential carrier of microorganisms. Airborne bacteria can attach to fine particles and be similarly transported through the air. It is important to note that not all particles are

aerosols, and not all aerosols contain microorganisms. However, particulate matter still provides useful indirect information about microbial dispersion dynamics.

Rather than relying on deterministic mathematical models, a probabilistic artificial intelligence approach was adopted. Classical analytical models often struggle to represent the complex and highly non-linear relationships between environmental conditions and bacterial concentration. These relationships are influenced by multiple interacting variables, such as human behavior, ventilation, and microclimatic conditions. Machine learning models are better suited to capture these interactions and to adapt to real-world variability. Several machine learning architectures were evaluated to identify the most suitable model for predicting the bacterial load from environmental measurements. Each candidate approach was selected to explore different modeling capabilities and trade-offs between complexity, interpretability, and expressive power. Support Vector Machines (SVMs) were first considered due to their strong theoretical foundations and their effectiveness in high-dimensional spaces. In this context, SVMs aim to identify an optimal hyperplane that minimizes the prediction error while maximizing the margin between data points. Through the use of kernel functions, they can implicitly project the input features into higher-dimensional spaces, allowing them to model non-linear relationships. However, despite this flexibility, SVMs tend to struggle when the underlying dependencies are highly complex or when the number of interacting variables increases, as is the case for environmental and biological processes. Decision Trees were also evaluated because of their intuitive structure and ease of interpretation. This class of models operates by recursively partitioning the feature space into regions that minimize intra-node variance. Each decision rule can be directly inspected, making the model's behavior transparent and easy to explain. Nevertheless, Decision Trees often suffer from limited generalization capability when used as standalone predictors. They are particularly sensitive to noise and tend to overfit the training data unless strong regularization or ensemble techniques are applied. Finally, a Multi-Layer Perceptron (MLP) was investigated to address the limitations observed in the previous approaches. MLPs are a class of feed-forward neural networks composed of multiple layers of interconnected neurons. Each neuron performs a weighted sum of its inputs followed by a non-linear activation function. This structure enables the network to learn hierarchical representations of the data and to approximate complex, non-linear functions. In the context of bacterial load prediction, the MLP demonstrated a superior ability to capture subtle interactions among environmental variables and temporal dynamics, ultimately providing more accurate and stable predictions than the other evaluated methods.

To determine the optimal network configuration, hyperparameter tuning was carried out using the Hyperband algorithm [259]. This method formulates hyperparameter optimization as an infinite-armed bandit problem, efficiently allocating computational resources to the most promising model configurations while discarding underperforming ones at an early stage. As a result of this automated search, the final architecture of the MLP consists of one input layer, followed by two fully connected hidden layers with five and three neurons, respectively, equipped with ReLU activation functions and two Dropout layers with a rate of 0.10 to prevent overfitting. The network terminates with a final fully connected layer with a single neuron, which outputs the predicted regression value.

A major challenge when applying deep learning to environmental microbiology is the limited availability of labeled data. In our experiments, ground-truth measurements were collected using SAS sampling. This procedure is time-consuming and costly, which restricted the dataset to about 160 validated samples. Such a small amount of data is not sufficient to train a deep neural network in a reliable way. To overcome this limitation, we introduced a data augmentation strategy aimed at increasing the size of the training set using synthetic samples. Different generative approaches were explored in order to progressively enlarge the dataset, reaching up to twelve times the size of the original data.

The simplest augmentation method was based on jittering. In this case, small Gaussian noise was added to the original feature vectors. The augmented data point x' is generated using the formula:

$$x' = x + \epsilon \tag{3.1}$$

where ϵ is a noise vector drawn from a standard normal distribution, with $\mu = 0$ and $\sigma = 0.03$. This produces new samples that are very close to the real measurements, while still introducing slight variations. As a result, the model becomes more robust to small fluctuations and noise in sensor readings.

To generate synthetic data with a more structured variability, autoencoders were also adopted. An autoencoder is a neural network that learns how to compress the input data into a lower-dimensional representation using an encoder model and then reconstruct it using a decoder. By learning this compact representation, the model captures the main structure of the environmental features, allowing for the generation of new samples that preserve the structural integrity of the original data. The specific architecture developed is a symmetric stack of Dense layers. The network compresses the input vector through a bottleneck and then expands it back to the original size. The input layer (matching the feature space size) is connected to a Dense layer of 5 neurons, which is further compressed into a latent bottleneck of 4 neurons. The reconstruction path mirrors the encoder, expanding from the 4-neuron latent space back to a Dense layer of 5 neurons, and finally to an output Dense layer matching the original input size.

This method was further advanced using Variational Autoencoders (VAEs). Unlike conventional autoencoders that learn a fixed representation for each input, VAEs model a probability distribution in the latent space. New samples are generated by drawing from this distribution, producing synthetic data that are not mere copies of the original measurements but represent statistically plausible variations consistent with the underlying environmental dynamics. A key element of the VAE is the sampling layer, which allows the model to propagate gradients through stochastic sampling. Rather than sampling the latent space directly, the network predicts the mean (z_{mean}) and the logarithm of the standard deviation (z_{log_sigma}) of the distribution. The latent vector z is then computed via the reparameterization trick:

$$z = z_{mean} + e^{z_{log\_sigma}} \cdot \epsilon$$

Where ϵ is a random noise vector drawn from a standard normal distribution with $\mu = 0$, and $\sigma = 0.1$.

Finally, Generative Adversarial Networks (GANs) were explored. These models rely on

the interaction between two neural networks trained simultaneously: a Generator, which creates synthetic samples from random noise, and a Discriminator, which evaluates whether a sample is real or generated. Through this adversarial process, the Generator incrementally refines its ability to produce realistic environmental measurements. Both networks employ a multilayer perceptron (MLP) architecture tailored to the tabular structure of the dataset. The Discriminator functions as a binary classifier, processing the environmental feature vector through a sequence of Dense layers to extract representations, ultimately producing a single output neuron representing the validity score. The Generator maps a random noise vector into the data space, and over successive training iterations, it progressively generates synthetic environmental readings that closely mimic real sensor data. This results in a high-fidelity augmentation of the training set.

3.2.2 Energy-Aware Data Transmission

The implementation of a continuous, real-time monitoring infrastructure for indoor safety poses significant operational challenges, particularly regarding energy autonomy. Since these IoT devices are often battery-powered and rely on wireless transmission technologies, the high frequency of data sampling and transmission required to track environmental dynamics can lead to rapid power depletion. To ensure that the monitoring of air quality and sanitization levels remains sustainable over long periods, it is crucial to optimize data communication, which is typically the most energy-intensive activity of an IoT device.

In this context, Prediction-Based Data Collection (PBDC) strategies emerge as an essential tool to balance monitoring accuracy with energy efficiency. These techniques exploit the temporal correlation inherent in environmental data to reduce redundant transmissions. Instead of a constant stream of data, the system only communicates when the measured values deviate significantly from a predicted model, thereby drastically extending the device's battery life. Building on this paradigm, two novel methodologies are introduced: Deep Learning-Based Data Collection (DLBDC) and Deep Learning-Driven Sensing (DLDS). Both approaches extend traditional PBDC by employing deep learning–based forecasting models deployed directly on edge devices. DLBDC leverages the predictive capabilities of these models to forecast future sensor values and dynamically determine when data transmission can be safely suppressed, thereby further reducing communication overhead compared to classical prediction models. In contrast, DLDS introduces an additional dimension of energy optimization by adopting a multi-step forecasting strategy. This approach enables the system to anticipate multiple future samples, allowing longer deep-sleep intervals and effectively reducing the frequency of node wake-ups without compromising the accuracy or continuity of environmental monitoring.

Problem formulation

The structure of IoT networks follows a distributed architectural paradigm in which many edge devices communicate with each other and with intermediary fog nodes before ultimately transmitting data to centralized cloud servers. Generally, an IoT monitoring task executed on an edge device can be modeled as a Finite State Machine (FSM) comprising distinct operational states, as reported in Figure 3.6. Specifically, for a task characterized

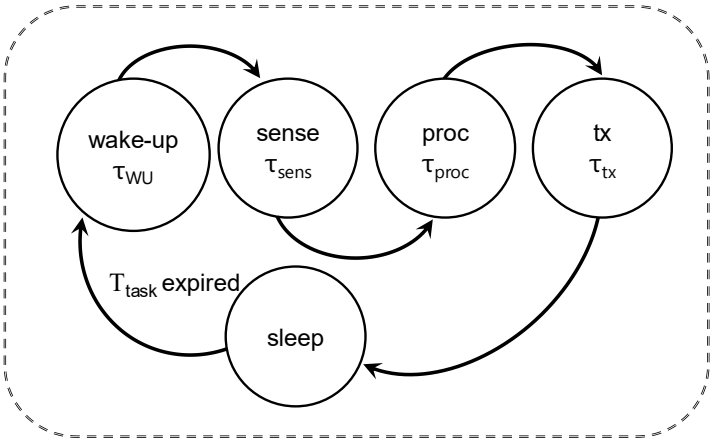


Figure 3.6: State diagram of a traditional IoT monitoring task.

by a period T_{task} representing the time interval between successive activations of the task, a sequence of five distinct states can be identified. When the timer expires, the IoT node transitions from a low-power sleep state to a transient state (*wake-up*), which involves the reactivation of the microcontroller unit and I/O peripherals. Once operational, the device initiates a measurement using its internal sensors (*sense*), potentially processes the data (*proc*), and subsequently transmits the information to the cloud (*tx*) before reverting to the *sleep* state. Each state is characterized by a specific energy consumption and duration, which are determined by the edge hardware platform. In general, the total energy consumption per task (E_{task}) can be expressed as the sum of the following components:

$$Etask = \mathcal{E}WU + \mathcal{E}sens + \mathcal{E}proc + \mathcal{E}tx + (Ttask - \tau_{WU} - \tau_{sens} - \tau_{proc} - \tau_{tx}) \cdot \mathcal{P}_{sleep} \tag{3.2}$$

Here, $\mathcal{E}_{WU}$, $\mathcal{E}_{sens}$, $\mathcal{E}_{proc}$, and $\mathcal{E}_{tx}$ represent the energy consumed during node wake-up, sensing, processing, and transmission, respectively (collectively referred to as *Active Energy*). The energy consumed in the sleep state is derived from the sleep power consumption $\mathcal{P}_{sleep}$ multiplied by the duration of the sleep interval. This interval is calculated as the task period T_{task} minus the cumulative time spent in active states (τ_{WU}, τ_{sens}, τ_{proc}, and τ_{tx}).

The energy consumption of IoT microcontrollers is highly dependent on their operational state. Active operation consumes considerably more power than low-energy modes, such as deep sleep or hibernation, which exploit idle states to minimize energy draw. Within active phases, wireless data transmission is typically the most power-intensive task, as it involves activating radio-frequency transceivers and associated signal-processing components, often accounting for the majority of the device's power usage. Consequently, reducing the frequency or volume of transmissions can yield substantial energy savings and improve the long-term sustainability of IoT deployments. Additionally, optimizing

the proportion of time spent in low-power states relative to active states provides another opportunity to lower energy consumption.

IoT monitoring tasks generally track parameters such as environmental conditions and can tolerate a degree of approximation. Unlike applications that require exact measurements for every sample, the system maintains integrity as long as reported values remain sufficiently close to the actual measurements. In this context, small deviations are acceptable provided they stay within predefined bounds [149]. The core challenge, therefore, is to balance data accuracy with energy efficiency, a fundamental requirement for effective real-time IoT monitoring.

Deep Learning-Based Data Collection

The DLBDC approach extends the traditional PBDC paradigm by integrating a Deep Learning-based forecasting model that runs simultaneously on edge devices and the cloud server. This setup allows the system to anticipate future sensor readings and transmit only the most relevant data, improving communication efficiency while maintaining the integrity and reliability of the information collected.

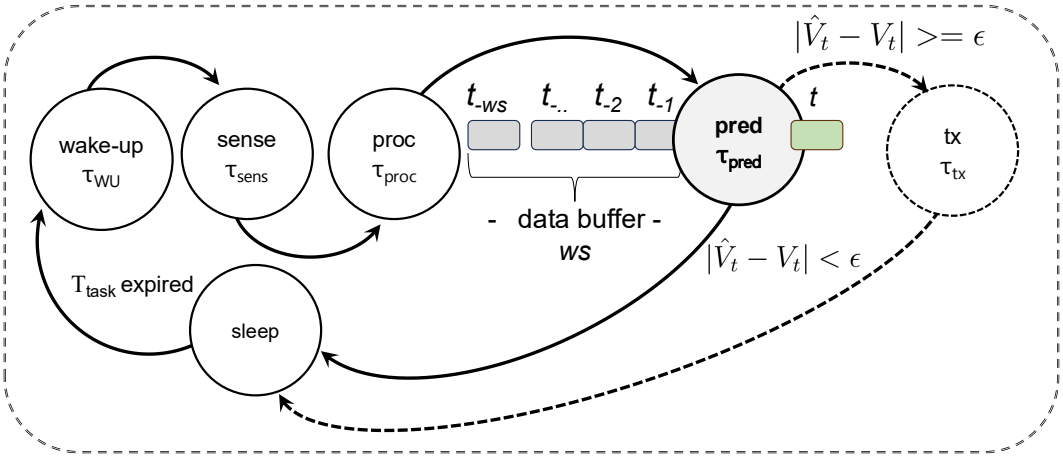


Figure 3.7: State diagram of the IoT monitoring task with the DLBDC methodology.

Figure 3.7 illustrates the FSM governing the DLBDC task. To better illustrate the methodology, we consider a scenario involving a single edge node executing a monitoring task characterized by a sampling period T_{task}. A deep-learning forecasting model is running on this task to predict the measured value. Let V_t denote the value measured by the device at time t, and $\hat{V}_t$ represent the corresponding predicted value at time t, derived using a historical buffer of length ws containing data sampled at times $[t_{-ws}, t_{-...}, t_{-2}, t_{-1}]$. Given a tolerance threshold ϵ accepted by the cloud server, a prediction is deemed valid if it satisfies the condition:

$$\hat{V}_t \in [V_t - \epsilon, V_t + \epsilon] \tag{3.3}$$

Specifically, if $|\hat{V}_t - V_t| < \epsilon$, the estimated value at time t is considered a sufficient approximation of the actual value. In this instance, data transmission is suppressed, and the cloud server independently predicts the value using the identical deep-learning model and substitutes $\hat{V}_t$ for V_t. This reduction in transmission is achievable because a shared forecasting model is employed, and synchronized historical data buffers are maintained on both the cloud and the edge device. This ensures the cloud's predictions mirror those of the edge node. On the other hand, transmission of the measured data is necessary only when the prediction error exceeds the tolerance range (i.e., $|\hat{V}_t - V_t| \geq \epsilon$).

In scenarios where the monitored parameter exhibits well-defined trends or gradual changes, the model will predict data with sufficient accuracy to remain within the acceptable range. For the sake of simplicity, we exclude potential issues related to unstable network connectivity, such as transmission errors and latency, assuming an ideal transmission medium, as these factors are outside the scope of this work. Generally, within the DLBDC framework, a transmission failure would be erroneously interpreted by the cloud server as the intentional suppression of a predictable sample (i.e., a value falling within the model's tolerance range). This would introduce minor inaccuracies in the historical record until the edge and cloud buffers are synchronized again. Regarding the forecasting models, it is important to note that they are trained offline on a dedicated machine and subsequently embedded into the device firmware. Therefore, periodic over-the-air transmission of the model itself is not required.

Building upon Equation 3.2, the energy model for the DLBDC approach is derived as follows:

$$\begin{aligned} \mathrm{E}_{task(DLBDC)} =& \mathcal{E}_{WU} + \mathcal{E}_{sens} + \mathcal{E}_{proc} + \mathcal{E}_{pred} + \mathcal{E}_{tx} \cdot (1 - SR_{tx}) + \\ & [\mathrm{T}_{task} - \tau_{WU} - \tau_{sens} - \tau_{proc} - \tau_{pred} - \tau_{tx} \cdot (1 - SR_{tx})] \cdot \mathcal{P}_{sleep} \end{aligned} \tag{3.4}$$

where SR_{tx} denotes the transmission suppression ratio achieved by the forecasting model for a specific tolerance ϵ. This ratio is dependent on the model's predictive capability. From an energy perspective, the contribution of the transmission energy $\mathcal{E}_{tx}$ is reduced by the factor SR_{tx}, while the computational cost of the prediction phase, $\mathcal{E}_{pred}$, is added to the task. Furthermore, regarding the sleep duration, the time required for prediction, τ_{pred}, must be subtracted (as the CPU remains active for inference), whereas the time allocated for data transmission, τ_{tx}, is reduced by the suppression ratio SR_{tx}.

Figure 3.8 illustrates the theoretical transmission energy savings achieved by the DLBDC strategy relative to a traditional IoT task, plotted against varying ratios of transmission cost to prediction cost ($\mathcal{E}_{tx}/\mathcal{E}_{pred}$) and different values of SR_{tx}. Assuming an energy ratio of approximately 10x, a typical value for many microcontrollers, a transmission suppression ratio of at least 30% is required to achieve significant savings (exceeding 20%). Naturally, as suppression increases, energy savings rise proportionally. However, the $\mathcal{E}_{tx}/\mathcal{E}_{pred}$ ratio is critical to the energy balance. For values below 10x, savings diminish rapidly, rendering the approach ineffective even at high suppression rates. This means that, given the fixed energy cost of transmission for a specific hardware platform, the predictive model must be carefully sized to ensure a favorable $\mathcal{E}_{tx}/\mathcal{E}_{pred}$ ratio. This presents a non-trivial optimization challenge, as more accurate models capable of achieving high

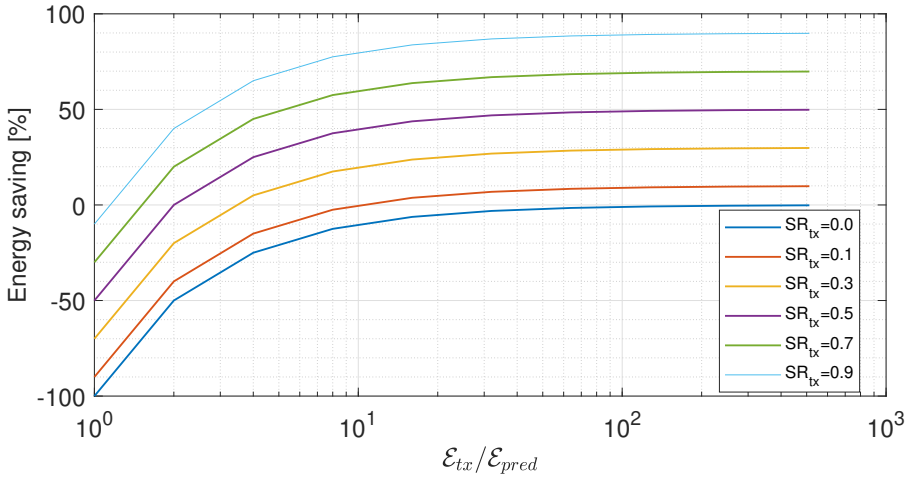


Figure 3.8: Theoretical transmission energy saved by the DLBDC approach when varying the $\mathcal{E}_{tx}/\mathcal{E}_{pred}$ ratio, for different values of SR_{tx}.

transmission suppression rates (SR_{tx}) typically entail greater computational complexity and, by extension, higher energy costs. Therefore, identifying the optimal trade-off between computational energy consumption and predictive performance is critical for overall system efficiency.

Finally, regarding the error introduced into the cloud-stored data, it is important to point out that the parameter ϵ acts as a strict upper bound, ensuring that no recorded data deviation exceeds this threshold.

Deep Learning-Driven Sensing

The proposed DLDS technique extends the classical PBDC paradigm by leveraging the capability of deep-learning multistep forecasting models to predict multiple future data points, specifically ts points, with high precision, utilizing a historical data buffer of length ws. Figure 3.9 illustrates the FSM of an IoT node implementing this technique. In this framework, at time t, the model predicts ts future points in a single inference step and schedules a wake-up event for a time $ts \times \mathrm{T_{task}}$ in the future before entering the sleep state.

Upon waking, the node updates the historical buffer by shifting the predicted values leftward. It then acquires a new measurement, V_t, and compares it with the most recent predicted value in the buffer, denoted as $\hat{V}_{t+ts}$ before the sleep interval, which now corresponds to $\hat{V}_t$. If $|\hat{V}_t - V_t| < \epsilon$, the final value of the estimated series is considered a valid approximation of the actual value. While a valid final approximation does not guarantee that all intermediate predictions fell within the tolerance threshold, the probability remains high given that prediction error in multistep forecasting models typically correlates with the temporal distance from the observation point.

In practice, if $\hat{V}_t$ is within tolerance, the node predicts the next ts future points and returns to the sleep state without transmitting data. Conversely, if the last predicted value

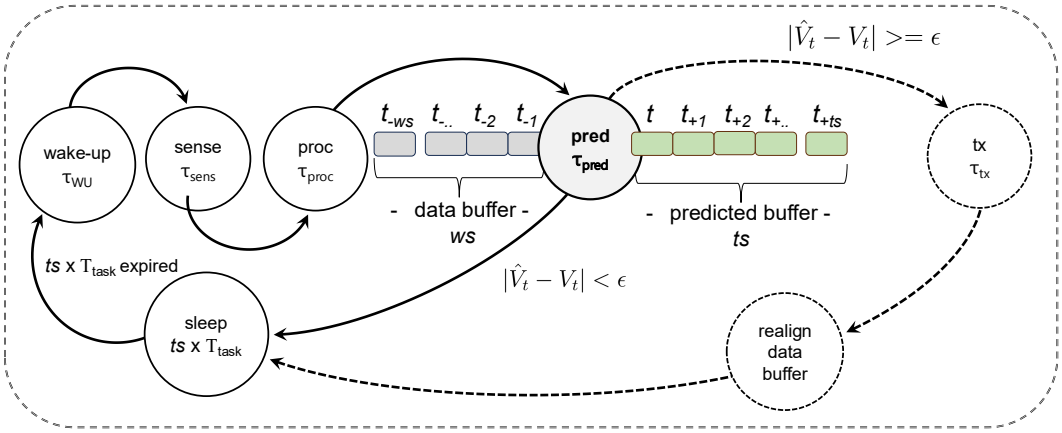


Figure 3.9: State diagram of the IoT monitoring task with the DLDS methodology.

exceeds the tolerance threshold, the measured value V_t must be transmitted to the cloud. This condition implies that the entire buffer has likely deviated beyond acceptable limits. Unfortunately, the intermediate real values are irrecoverable, as they were never acquired in the first place. In such cases, the only viable strategy is to *realign the buffer* with new data to synchronize the model with the actual distribution, thereby enhancing future prediction accuracy.

The realignment methodologies may range from completely refilling the buffer with new values to immediately resuming forecasting using the latest measured data point. We adopt an intermediate strategy. Using the newly acquired value $\hat{V}_t$, we perform a linear interpolation over the existing buffer values to correct the overall trend. This approach avoids keeping the node active for ws cycles to fully repopulate the buffer and prevents the introduction of abrupt discontinuities associated with simply appending the latest value. From an energy perspective, the realignment contribution is negligible given its reduced computational complexity.

Overall, the energy consumption of the task is heavily influenced by the value of ts, which determines the sleep duration and, consequently, the activation frequency. For instance, for a task with period equal to $\mathrm{T_{task}}$, the active states occur only $\mathrm{T_{task}}/ts$ times. Therefore, the energy model for the DLDS approach is defined as:

$$\begin{aligned}\mathrm{E}_{task(DLDS)} =& \frac{\mathcal{E}_{WU} + \mathcal{E}_{sens} + \mathcal{E}_{proc} + \mathcal{E}_{pred} + (\mathcal{E}_{tx} + \mathcal{E}_{re}) \cdot (1 - SR_{tx})}{ts} + \\ & [\mathrm{T}_{task} - \frac{\tau_{WU} + \tau_{sens} + \tau_{proc} + \tau_{pred} + (\tau_{tx} + \tau_{re}) \cdot (1 - SR_{tx})}{ts}] \cdot \mathcal{P}_{sleep}\end{aligned} \tag{3.5}$$

where $\mathcal{E}_{re}$ and τ_{re} represent the energy consumption and the duration of the *realign* state, respectively. Moreover, neglecting the contribution of the realignment state, the equation

simplifies to:

$$\begin{aligned} \mathrm{E}_{task(DLDS)} = & \frac{\mathcal{E}_{WU} + \mathcal{E}_{sens} + \mathcal{E}_{proc} + \mathcal{E}_{pred} + \mathcal{E}_{tx} \cdot (1 - SR_{tx})}{ts} + \\ & [\mathrm{T}_{task} - \frac{\tau_{WU} + \tau_{sens} + \tau_{proc} + \tau_{pred} + \tau_{tx} \cdot (1 - SR_{tx})}{ts}] \cdot \mathcal{P}_{sleep} \end{aligned} \tag{3.6}$$

Notably, for $ts = 1$, this model is identical to the DLBDC methodology described by Equation 3.4.

Figure 3.10 illustrates the overall energy savings achieved by the DLDS strategy compared to a traditional IoT task, as a function of the number of predicted samples (ts) and varying task periods (T_{task}). The plot is derived from Equation 3.6, assuming active task power consumption is 100 times greater than the sleep state, $\mathcal{E}_{tx}/\mathcal{E}_{pred} = 10$, and $SR_{tx} = 0.7$. Interestingly, the majority of energy savings are achieved by predicting just two or three future samples, regardless of T_{task}. This observation is critical, as reconstruction error typically increases with the number of consecutively predicted points. Consequently, extending the prediction horizon too far into the future yields diminishing returns and may compromise data fidelity. The extent of energy savings is also influenced by the task period; in scenarios with extended sleep durations, the energy contribution of active states becomes increasingly negligible, limiting the potential for further reduction.

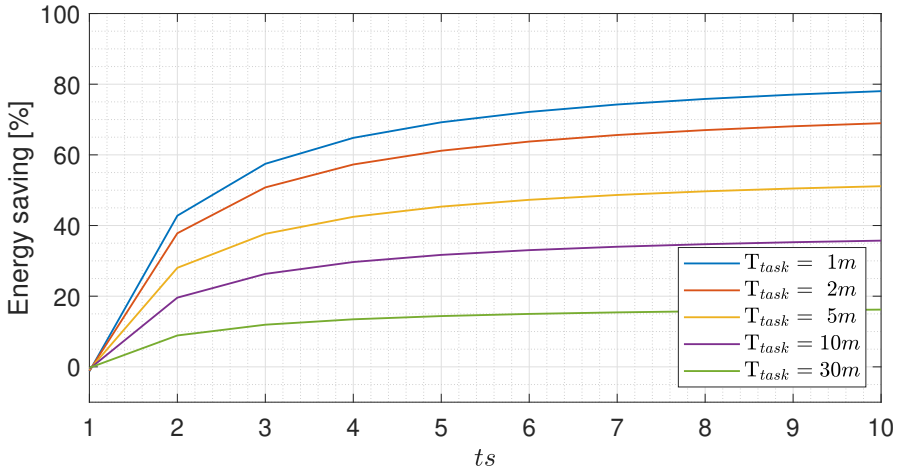


Figure 3.10: Theoretical overall energy saved by DLDS approach when varying the number of predicted samples (ts), for different values of T_{task}.

Finally, it is important to note that, unlike the DLBDC methodology, the ϵ parameter in DLDS does not establish a strict upper bound on the actual data reconstruction error, due to the lack of knowledge of the error on unacquired values. Therefore, careful tuning of model parameters, specifically the prediction horizon (ts), is crucial to ensuring both predictive accuracy and reliable data reconstruction.

Algorithms implementation

Algorithm 1 presents the pseudocode for the DLBDC procedure as executed on the edge node. The algorithm takes as input the tolerance threshold ϵ, which specifies the allowable error margin for the monitoring task. Internally, it maintains a circular buffer of size ws to store the most recent measurements, along with the predicted value $\hat{V}_t$ computed during the previous iteration.

Algorithm 1 Pseudo-code of the DLBDC algorithm running on the edge node.

Ensure: buffer ▷ Buffer of size ws
Ensure: $\hat{V}_t$ ▷ Predicted value at time t
Require: ϵ ▷ Tolerance threshold
1: **while** true **do**
2: $V_t \leftarrow measureSensor()$
3: **if** $|\hat{V}_t - V_t| < \epsilon$ **then**
4: // No need to send
5: $updateBuffer(buffer, \hat{V}_t)$
6: **else**
7: $sendToServer(V_t)$
8: $updateBuffer(buffer, V_t)$
9: **end if**
10: $\hat{V}_t \leftarrow predict(buffer)$
11: $deepSleep(T_{task})$
12: **end while**

In each iteration, the edge node samples its sensor to obtain V_t. If the deviation between this value and the predicted value $\hat{V}_t$ exceeds the tolerance ϵ, the actual measurement is transmitted to the server; otherwise, transmission is suppressed. Regardless of the transmission status, the circular buffer is updated via the `updateBuffer` function, which operates on a First-In-First-Out (FIFO) basis, removing the oldest entry and appending either V_t or $\hat{V}_t$. Subsequently, a new prediction $\hat{V}_t$ is computed based on the updated buffer. Upon completion of these operations, the device transitions into a deep-sleep state until the next scheduled iteration.

Algorithm 2 presents the pseudocode for the DLDS technique. Like DLBDC, this algorithm requires the tolerance threshold ϵ and keeps a circular buffer of size ws for storing recent measurements. A distinguishing feature is that the internal state includes a list $\hat{\overline{V}}$ of size ts, containing the sequence of values predicted during the previous iteration. The value used for comparison in the algorithm corresponds to the last element of this list, denoted as $\hat{V}$.

In each iteration, the node samples the sensor to obtain V_t and verifies whether the deviation from $\hat{V}$ exceeds the tolerance ϵ. If the difference lies within the acceptable range (i.e., $|\hat{V} - V_t| < \epsilon$), transmission is suppressed, and the entire sequence of predicted values $\hat{\overline{V}}$ is integrated into the circular buffer. Conversely, if the error exceeds the threshold, the actual measured value V_t is transmitted to the server, and a realignment strategy is invoked via the `realignBuffer` function. Lastly, the updated buffer is utilized to predict the next sequence $\hat{\overline{V}}$, after which the node enters deep sleep for a duration of $ts \times T_{task}$

Algorithm 2 Pseudo-code of the DLDS algorithm running on the edge node.

Ensure: buffer ▷ Circular buffer of size ws
Ensure: $\hat{\overline{V}}$ ▷ List of predicted values with size ts
Require: ϵ ▷ Tolerance threshold

while true **do**
  $V_t \leftarrow measureSensor()$
  if $|\hat{V} - V_t| < \epsilon$ **then**
    // No need to send
    $updateBuffer(buffer, \hat{\overline{V}})$
  else
    $sendToServer(V_t)$
    $updateBuffer(buffer, \hat{\overline{V}})$
    $realignBuffer(buffer, V_t)$
  end if
  $\hat{\overline{V}} \leftarrow predict(buffer)$
  $deepSleep(ts \times T_{task})$
end while

iterations.

For buffer realignment, we implemented a linear interpolation strategy. This approach approximates intermediate buffer values by assuming a linear progression between the oldest element (i.e., the first) and the most recent element (i.e., the last). First, the slope (Δ) of the line is computed by subtracting the value of the first point from the last, divided by the number of intervening intervals. Subsequently, the value for each position in the buffer is calculated as $y_i = y_{first} + i \cdot \Delta$, where y_{first} is the oldest element and i represents the index offset from the start.

Both DLBDC and DLDS algorithms are deliberately designed to be lightweight in terms of implementation and computational overhead. The most resource-intensive tasks, updating and realigning buffers, are handled using simple linear interpolation and basic list operations on small data sets. This streamlined design makes the algorithms well-suited for deployment on resource-constrained edge devices, ensuring low latency and minimal energy consumption.

The proposed DLBDC and DLDS algorithms were characterized and validated using a dedicated simulation framework. This framework accepts as input various datasets containing real-world time-series data collected in indoor environments. This setup enables a direct comparison with traditional PBDC techniques relying on classical modeling approaches, highlighting the specific advantages and limitations of data-driven solutions under comparable scenarios.

Finally, we considered a case study utilizing three popular IoT platforms, wherein the algorithms were deployed to empirically quantify energy consumption on resource-constrained hardware. The complete source code for the simulation framework, including the implementations of both algorithms and all scripts for data preprocessing, model training, and experimental evaluation, is publicly available at: `https://github.com/IoTUniurb/deep-learning-based-data-collection`.

Forecasting Models

The core component of the proposed sensing framework is a Deep Learning-based forecasting model. In order to evaluate the approach across different technologies, we designed three distinct model architectures: Model #1, Model #2, and Model #3. Figure 3.11 shows a diagram of the different models, while Table 3.1 provides a summary of their respective hyperparameter configurations.

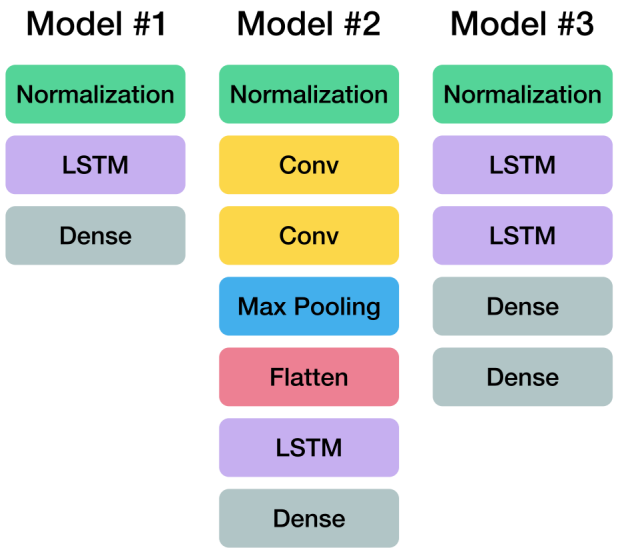


Figure 3.11: Diagram highlighting the architecture of the three forecasting models.

Model #1 is composed of a single Long Short-Term Memory (LSTM) layer with 10 units. Model #2 uses a hybrid structure, combining two Convolutional Neural Network (CNN) layers with 64 filters each, followed by one LSTM layer of 10 units. Model #3 consists of two stacked LSTM layers, each with 10 units, followed by a dense layer of 30 neurons. All three architectures conclude with a final dense layer, sized according to the technique: one output neuron for DLBDC, and ts neurons for DLDS.

Table 3.1: Hyperparameter configuration of the three forecasting models.

ID	Net Type	Net Structure	Params [k]	Net Size [kB]
#1	1LSTM_1D	$10U\_tsD$	≈ 0.5	≈ 2
#2	2CNN_1LSTM_1D	$64F\_64F\_10U\_tsD$	≈ 14	≈ 55
#3	2LSTM_2D	$10U\_10U\_30D\_tsD$	≈ 1.7	≈ 7

All three models are designed to be lightweight and compact. Model #2 is the most complex, with roughly 15,000 parameters and an estimated storage footprint of 55 kB. Model #1 is the smallest, comprising about 500 parameters and occupying around 2 kB.

Model #3 strikes a balance between size and capability, containing approximately 1,700 parameters and requiring close to 7 kB. These figures are approximate, as the exact parameter count depends slightly on the size of the final dense layer (determined by ts), but the variation is minimal.

Training was carried out on various environmental datasets using a supervised approach. Input data were segmented into consecutive windows of size ws, with the following ts values serving as ground truth for prediction. For the DLBDC technique, ts was set to 1. This windowing converts the time series into input/output pairs, supporting effective training and improving model performance. During training, windows were grouped into batches of size 32 to streamline computation and reduce epoch duration.

3.3 Experimental Setup

3.3.1 Indoor Environmental Datasets

Indoor air quality is a key factor in the management of indoor environments, with direct implications for occupants' health and comfort. From a methodological standpoint, it also represents a challenging domain for data-driven modeling, as indoor environmental variables exhibit heterogeneous temporal dynamics and complex correlations. These challenges make it well-suited for validating the proposed deep learning models and the PBDC algorithms. To support this evaluation, an ad hoc dataset was constructed by monitoring a heterogeneous set of environmental parameters selected to serve two complementary objectives: providing indirect proxies of airborne bacterial load and stressing the forecasting algorithms through diverse statistical behaviors. Data were collected by deploying sensor nodes within the classrooms of the Computer Science and Technology degree program at the University of Urbino, Italy. We selected this specific environment because university classrooms provide a reliable and challenging benchmark for indoor monitoring. They exhibit highly dynamic behavior, characterized by periods of high occupancy and rapid environmental changes during lectures, alternating with periods of inactivity during nights, weekends, and holidays. This duality closely resembles the typical dynamics of smart building applications.

The IoT nodes were deployed in three specific classrooms located on the second floor of Collegio Raffaello, a historical building dating back to the 18th century situated in the city center of Urbino. The monitored rooms are named after prominent figures in computer science: Turing, Olivetti, and Von Neumann. Figure 3.12 shows the floor plan of the building's second floor, with the monitored areas highlighted in orange. The environmental conditions in these rooms are strictly correlated with the daily academic routine. The building follows a rigid schedule. At 07:30, the building guardian opens the main entrances, allowing access to faculty and staff. Typically, the most significant influx of students occurs between 08:30 and 09:00. From 09:00 to 13:00, the first block of lectures takes place. Then, from 13:00 to 14:0,0 there is the lunch break, during which occupancy drops significantly. Finally, from 14:00 to 18:,00 the afternoon sessions resume, with a second peak of arrivals occurring shortly before 14:00. This routine is consistent during the academic semesters (February–June and September–December). The data

Figure 3.12: Floor plan of the second floor of Collegio Raffaello. The monitored classrooms are highlighted in orange.

acquisition campaign spanned two full calendar years, from January 1, 2023, to December 31, 2024. During this period, the sensors were configured to wake up and sample the environment with a frequency of 5 minutes. The complete dataset, along with the scripts used for statistical analysis and plotting, has been made publicly available at: `https://github.com/IoTUniurb/classroom_dataset_scripts`

The environmental variables included in the dataset were selected to capture complementary aspects of indoor occupancy, air quality, and human activity, while providing signals with greatly different statistical properties for algorithmic validation. Rather than directly measuring biological contamination, which would require invasive and impractical instrumentation, the adopted approach relies on a set of physical and chemical proxies that are strongly influenced by human presence and behavior and are commonly used in indoor monitoring studies. Carbon dioxide (CO_2) concentration was collected as a primary indicator of human occupancy and ventilation effectiveness. CO_2 is a direct byproduct of human respiration and therefore shares the same production source as human-borne microorganisms. However, unlike biological agents, it behaves as a conservative gas that rapidly mixes within the room volume and does not settle over time. For this reason, while CO_2 alone cannot represent bacterial load, it provides a reliable and smoothly varying proxy of cumulative human presence and air renewal conditions. In the dataset, this quantity was measured using the MH-Z19B, a Non-Dispersive Infrared (NDIR) sensor that estimates gas concentration by measuring infrared absorption at wavelengths specific to CO_2, a well-established and stable technique for indoor monitoring [260].

To complement this information and capture more transient aspects of human emissions, the dataset includes measurements of total volatile organic compounds (TVOCs). These compounds are released during respiration and speech and tend to follow similar temporal trends to CO_2, but with significantly shorter persistence in the air due to their chemical reactivity with oxygen. As a result, TVOCs provide additional insight into the freshness of emissions and short-term occupancy dynamics. TVOC concentrations were acquired using the SGP30, a multi-pixel metal-oxide gas sensor capable of estimating overall VOC levels and providing an equivalent CO_2 signal, thus extending the chemical characterization of indoor air [261]. Closely linked to the behavior of bio-aerosols are temperature and relative humidity, which were monitored throughout the data collection period. These thermodynamic variables play a crucial role in determining the physical evolution of respiratory droplets. Temperature affects evaporation rates, while humidity influences whether droplets remain suspended in the air or precipitate onto surfaces. Both parameters were measured using the DHT11 sensor module, which integrates a resistive humidity sensing element and a thermistor to provide low-cost but continuous monitoring of indoor environmental conditions [262]. To directly observe the physical carriers that may transport microorganisms, the dataset includes measurements of particulate matter, specifically focusing on PM_{10} and $PM_{2.5}$. The coarser particles can act as potential carriers for microorganisms, while finer particles, with diameters below 2.5 μm, are of particular concern due to their ability to penetrate deeply into the respiratory system. Although not all particulate matter contains biological material, its concentration provides valuable information on aerosol dynamics and human-induced resuspension. These measurements were obtained using the SPS30, an optical sensor based on laser-scattering technology that

delivers high-precision mass concentration estimates across multiple particle size classes [263]. Acoustic noise levels were monitored as a direct proxy of human activity, particularly relevant in educational environments where occupancy patterns are strongly reflected in sound dynamics. Noise signals are characterized by rapid fluctuations and high variance, making them fundamentally different from slowly varying environmental variables. Acoustic data were collected using the INMP441, a digital MEMS microphone that continuously samples the ambient sound field [264]. Finally, background gamma radiation (γ *dose rate*) was included as an intentionally adversarial variable for predictive modeling. Measured through a Geiger–Müller tube interface, this signal is derived from discrete photon counts that follow a Poisson distribution, exhibiting high intrinsic stochasticity and weak temporal correlation. While not directly related to indoor air quality, its inclusion provides an extreme test case for compression, forecasting, and decision-making algorithms operating under uncertainty. Together, these measurements form a heterogeneous dataset that captures both occupancy-driven phenomena and diverse statistical behaviors, providing a comprehensive and challenging benchmark for the validation of the proposed models and methodologies.

3.3.2 State-of-the-art Comparison Techniques

To rigorously evaluate the performance of the proposed DLBDC and DLDS methodologies, we established a comparison baseline using state-of-the-art PBDC techniques grounded in traditional statistical approaches. Specifically, we selected two widely adopted algorithms characterized by their low computational complexity and suitability for resource-constrained devices: Derivative-Based Prediction (DBP) and the Kalman Filter.

The **Derivative-Based Prediction (DBP)** algorithm, initially introduced by Raza et al. [265, 202], is a lightweight forecasting technique developed for Wireless Sensor Networks (WSNs). Its central idea is that many environmental variables change gradually over time, allowing their short-term behavior to be effectively modeled using a linear approximation. Unlike regression methods that minimize the mean squared error across multiple points, DBP emphasizes alignment with the observed trend, making it particularly robust against high-frequency noise and outliers commonly encountered in low-cost sensors.

The operation of the algorithm is divided into two distinct phases. During model generation, the sensor node collects a fixed sequence of m data points, known as the *learning window*. To compute a robust trend line while filtering out noise, the algorithm identifies the first and the last l samples of this window as *edge points*. The slope of the prediction model, denoted as Δ, is calculated based on the difference between the averages of these edge points. This operation is analogous to computing a numerical derivative over the window, hence the name "Derivative-Based".

Once initialized, the node continuously monitors incoming data, maintaining a sliding window of recent samples to ensure the model remains up-to-date. For each new measurement, the algorithm predicts the expected value using the calculated slope Δ. Model validity is assessed using two thresholds: the value tolerance (ϵ_V), defining the maximum permissible deviation between the prediction and the actual measurement, and the time tolerance (ϵ_T), specifying how many consecutive violations of ϵ_V are allowed before the

model is considered invalid. While the prediction error remains within these limits, the model is regarded as valid, and data transmission is suppressed. When the error exceeds the thresholds, reflecting a meaningful shift in the observed trend, a new model is computed from the current window, and the updated slope Δ is transmitted to the central server.

The **Kalman Filter (KF)** is a recursive estimator widely regarded as the optimal solution for inferring the hidden state of a linear dynamic system subject to Gaussian noise. Its recursive formulation is highly efficient for embedded applications, requiring only the current state estimate and associated uncertainty to generate the next prediction, without storing extensive historical data.

The filter operates on a state-space representation of the system. Its execution loop is divided into two distinct phases: (i) prediction phase, and (ii) update phase. In the prediction phase, the filter projects the current state estimate forward in time to produce a *a priori* estimate. The system evolution is modeled as:

$$x_t = Fx_{t-1} + \omega_t \tag{3.7}$$

$$P_t = FP_{t-1}F^T + Q_t \tag{3.8}$$

Here, x_{t-1} represents the a priori state estimate at the previous time step, and P_t is the corresponding error covariance matrix, representing the uncertainty of the prediction. F is the state-transition matrix that defines how the system evolves physically, while Q_t represents the process noise covariance (modeling the inherent unpredictability of the system).

Once a new physical measurement z_t is acquired, the algorithm enters the update phase. First, the Kalman gain (K) is computed to determine the optimal weight between the model's prediction and the actual measurement:

$$K = P_tH^T(HP_tH^T + R)^{-1} \tag{3.9}$$

where H is the observation model mapping the state space to the measurement space, and R is the measurement noise covariance. Finally, the state estimate and its uncertainty are refined (*a posteriori*) using the measurement residual:

$$x_t = x_t + K(z_t - Hx_t) \tag{3.10}$$

$$P_t = (I - KH)P_t \tag{3.11}$$

In the context of PBDC, we implemented the Dual Kalman Filter (DKF) strategy proposed by Jain et al. [204]. This approach leverages the deterministic nature of the prediction phase to synchronize the sensor and the cloud without continuous communication. For each data stream i, two identical Kalman Filters are instantiated: one running locally on the IoT node (KF^i_{remote}) and a synchronized copy running on the central server (KF^i_{server}). At the sensor, the node computes the prediction $z_t = Hx_{t-1} + v_t$. It then compares this prediction with the actual sensor reading. If the error falls below a predefined tolerance threshold, the measurement is discarded, and no transmission occurs. If no

data packet is received, the server assumes the prediction was accurate and uses its local KF^i_{server} estimate as the valid data point. If a packet is received (meaning the prediction failed), the server updates its filter with the new real value to resynchronize the state. This dual architecture significantly reduces radio usage while maintaining a coherent time series on the backend. Additionally, the system responsiveness can be tuned via the transition parameter F_i, allowing the filter to adapt to signals with different dynamics (i.e., slow-moving temperature vs. fast-changing noise).

3.3.3 Forecasting Accuracy Metrics

In order to validate the forecasting models during the training and testing phases, and to quantify the performance achieved by the proposed DLBDC and DLDS techniques, we employ two standard statistical metrics capable of measuring the reconstruction error introduced by the models: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

The MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It computes the average absolute deviation between the predicted and observed values, expressed in the same physical units as the original data (i.e., °C for temperature or ppm for CO_2). It is mathematically defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.12}$$

where N is the total number of samples, y_i represents the ground truth value actually measured by the sensor at time step i, and $\hat{y}_i$ represents the value predicted by the forecasting model for the same time step.

While MAE provides an intuitive measure of error in absolute terms, it makes it difficult to compare performance across different variables with vastly different scales (i.e., comparing temperature errors around 20 with CO_2 errors around 400). To address this, we utilize the MAPE, which expresses the prediction accuracy as a percentage relative to the true value. It is defined as:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}| \tag{3.13}$$

The selection of these two specific metrics provides a holistic view of model performance. MAE offers a direct interpretation of the physical error, which is crucial for assessing whether the reconstruction meets the specific application requirements. Conversely, MAPE allows for scale-independent comparisons between different sensor types. However, it is important to note that MAPE penalizes errors on small true values more heavily than on large ones and can become unstable if y_i approaches zero. Therefore, it is interpreted in conjunction with the MAE to ensure a robust evaluation.

3.3.4 Simulation and Training Architectures

A comprehensive experimental framework was established to rigorously validate the proposed PBDC architectures, encompassing data preprocessing, model training on high-performance hardware, optimization for embedded deployment, and a realistic simulation environment. The environmental dataset was first partitioned into two distinct segments: a training set and a simulation set. The simulation segment was strictly excluded from the training phase, ensuring that the algorithms were evaluated solely on unseen data, thereby providing an unbiased assessment of their generalization capabilities. For the training phase, the time series data were reshaped to fit a supervised learning paradigm. The continuous stream of sensor readings was transformed into a sequence of input-output pairs using a sliding window approach. Specifically, the input consists of a sequence of ws consecutive samples (the observation window), while the target output comprises the immediately subsequent ts values (the prediction horizon). To maximize the volume of training examples and capture fine-grained temporal dynamics, these windows were generated with a stride of one time step. This procedure allows the neural networks to learn the complex temporal dependencies required to forecast future environmental states based on historical trends.

The training of the forecasting models was executed using the open-source deep learning library Keras [266] running on top of TensorFlow [267] (versions 2.15). The computational workload was handled by a high-performance workstation equipped with dual Intel® Xeon® Silver 4314 CPUs and accelerated by three NVIDIA® A100 GPUs. However, the ultimate goal of this research is to deploy these models on resource-constrained IoT nodes. To bridge the gap between the high-performance training environment and the microcontroller target, first, the trained models were converted into a hardware-friendly format using LiteRT [268] (formerly TensorFlow Lite). This step optimizes the graph structure and quantizes weights to reduce model size and latency. Then, the LiteRT for Microcontrollers [269] framework translates the optimized models into statically allocated C byte-arrays. This representation allows the models to be directly embedded into the firmware and executed without the overhead of an operating system or dynamic memory allocation.

The simulation framework was developed in Python version 3.10 and executed on the same high-performance workstation. For this phase, we used only the data belonging to the simulation set, which was reserved exclusively for this purpose. This choice ensures that all algorithms are evaluated on previously unseen data, providing a realistic assessment of their generalization performance. In contrast to the training phase, which relies on supervised learning and batch processing, the simulation is designed to closely replicate real operating conditions. The framework processes data sequentially, one sample at a time, while managing a circular buffer of size ws to emulate the memory constraints of the microcontroller. Exact software replicas of the DLBDC and DLDS approaches were implemented, following Algorithms 1 and 2. In addition, the two baseline techniques used for comparison, introduced in Section 3.3.2, were also implemented within the same simulation environment to ensure a fair and consistent evaluation. Within this setup, the `measureSensor` function was emulated by reading the next value from the dataset

at each time step. To isolate the algorithmic behavior from hardware-dependent effects, network transmissions and deep-sleep operations were abstracted and treated as logical events. These events were logged to compute relevant performance indicators, such as energy consumption and data reduction rates, without introducing platform-specific biases. During execution, all operations performed by the algorithms were recorded to enable detailed analysis. Finally, fixed random seeds were used for all stochastic components to guarantee the reproducibility of the simulation results.

3.3.5 Hardware Platforms

An ad-hoc monitoring task is implemented on three distinct hardware platforms to evaluate the energy savings and generalizability of the proposed methodology. These devices were selected to represent different segments of the current IoT landscape: the ESP32 (representing the standard, low-cost market), the Raspberry Pi Pico 2 W (representing the latest generation of dual-architecture microcontrollers), and the NXP MCXN947 (representing industrial-grade edge AI devices).

The **ESP32** is an MCU developed by Espressif Systems. It is widely regarded as the industry standard for low-cost IoT applications due to its robust integration of wireless connectivity and processing power [270]. Specifically, we utilized the ESP32-WROOM-32 DevKit module, which is built around a highly integrated System-on-Chip (SoC) featuring a dual-core Xtensa® 32-bit LX6 microprocessor running at 240 MHz. A key feature of this platform is its integrated wireless connectivity, supporting both 2.4 GHz Wi-Fi and Bluetooth. This dual connectivity capability allows the device to easily interface with gateways and mobile devices without the need for external radio modules. The device is equipped with approximately 520 KB of internal SRAM and 4 MB of external flash memory for firmware storage. From an architectural standpoint, the ESP32 is designed to support concurrent real-time tasks. It exposes a rich set of peripheral interfaces through its 48 General-Purpose Input/Output (GPIO) pins, including Analog-to-Digital Converters (ADCs), SPI, UART, and I2C buses, which facilitate the integration of heterogeneous sensors. Firmware development was carried out using the Espressif IoT Development Framework (ESP-IDF) [257]. This C/C++ framework provides a comprehensive software stack, offering fine-grained control over hardware features, such as deep-sleep and hibernation modes, essential for optimizing power consumption in battery-operated scenarios.

The **Raspberry Pi Pico 2 W** is a compact and versatile development board designed by the Raspberry Pi Foundation for embedded and IoT applications. This platform is based on the RP2350 microcontroller, which introduces an uncommon dual-core, dual-architecture design [271]. The device allows developers to choose between two Arm® Cortex-M33 cores or two open-standard RISC-V Hazard3 cores, both operating at clock frequencies of up to 150 MHz. The microcontroller provides 520 KB of on-chip SRAM, complemented by 4 MB of external flash memory, offering sufficient resources for moderately complex embedded workloads. From a connectivity perspective, the Pico 2 W integrates a 2.4 GHz wireless module supporting Wi-Fi 802.11n and Bluetooth 5.2, features that are particularly relevant for low-power IoT deployments. The board exposes 40 GPIO pins, enabling support for a wide range of peripherals, including a 12-bit analog-to-digital

converter and multiple Programmable I/O (PIO) state machines. Software development is supported through the official Pico SDK for C/C++ [272], as well as through MicroPython [273], making the platform suitable both for rapid prototyping and for more performance-oriented embedded development.

The **MCXN947** is a compact, scalable development board from NXP [274], tailored for rapid prototyping of secure, intelligent industrial IoT solutions. It features a dual-core Arm® Cortex-M33 MCU operating at 150 MHz, with up to 2 MB of dual-bank flash memory and 512 KB of full-ECC RAM, providing high reliability for data-intensive applications. A notable aspect of the MCXN947 is its dedicated hardware acceleration. The board includes an eIQ® Neutron N1-16 Neural Processing Unit (NPU) and a Digital Signal Processing (DSP) co-processor, allowing advanced machine learning tasks to run efficiently at the edge while offloading computation from the main CPU. Connectivity is comprehensive, including Ethernet, and the platform supports expansion via 124 GPIO pins. Firmware development is carried out using the MCUXpresso IDE [275], facilitating streamlined deployment of embedded applications.

To accurately quantify the energy impact of the proposed algorithms, the devices under test were powered using an NGMO2 Rohde & Schwarz dual-channel power supply [276], configured to provide a stable voltage appropriate for each board. To measure the current consumption with high temporal resolution, a shunt resistor of 2.5 Ω was placed in series with the power line. The voltage drop across the resistor was continuously sampled using a National Instruments NI-DAQmx PCI-6251 16-channel data acquisition board, connected via a BNC-2120 shielded connector block [277, 278]. This high-fidelity acquisition system allowed us to isolate and integrate the power consumption profile of every distinct phase of the firmware execution.

Table 3.2 summarizes the average power ($\mathcal{P}$), duration (τ), and total energy expenditure ($\mathcal{E}$) measured for each operational state across the three evaluated boards. Since the computational load of the prediction phase depends on the algorithmic complexity, we characterized it separately for the proposed Deep Learning models ($Pred_{DL}$), as well as for the benchmark techniques ($Pred_{DBP}$, and $Pred_{KF}$).

Table 3.2: Characterization of power and energy consumption of the different operation states for the three representative IoT devices (ESP32, Pico 2 W, and MCXN947).

	ESP32			Pico 2 W			MCXN947		
State	$\mathcal{P}$ **[mW]**	τ **[ms]**	$\mathcal{E}$ **[mJ]**	$\mathcal{P}$ **[mW]**	τ **[ms]**	$\mathcal{E}$ **[mJ]**	$\mathcal{P}$ **[mW]**	τ **[ms]**	$\mathcal{E}$ **[mJ]**
$WakeUp$	155	2775	430	68	5800	394	224	3262	731
$Sense$	375	54	20	68	82	6	414	39	16
$Proc.$	387	4	2	77	6	1	431	3	1
$Pred_{DL}$	387	61	24	77	226	17	431	42	18
$Pred_{DBP}$	387	30	12	77	131	10	431	19	8
$Pred_{KF}$	387	120	46	77	426	33	431	88	38
Tx	469	690	324	576	603	347	418	125	52
$Sleep$	29	-	-	11	-	-	31	-	-

The $WakeUp$ state is the most energy-demanding across all evaluated devices. Its high impact derives primarily from its relatively long duration, which can reach up to 5.8 seconds on the Pico 2 W. When resuming from deep sleep, each platform must execute a complete boot sequence, including system initialization, sensor configuration, and network

setup. Although the Pico 2 W features a low power draw, its reduced operating frequency leads to a prolonged wake-up phase. In contrast, the MCXN947 achieves the fastest wake-up but at the highest energy cost, peaking at 731 mJ. The ESP32 shows an intermediate behavior, combining a short wake-up time with a moderate overall energy expenditure.

The sensing (*Sense*) and processing (*Proc*) phases account for only a minor fraction of the total energy consumption across all platforms, primarily because of their brief duration. Among the evaluated boards, the ESP32 shows the highest energy usage during these stages, around 20 mJ for sensing and 2 mJ for processing, yet their contribution remains minimal when compared to the device's overall energy profile.

The prediction phase (*Pred*), associated with the different forecasting strategies, introduces a moderate and well-contained energy overhead. On the ESP32, the proposed deep learning models consume, on average, about 24 mJ per inference, compared to 12 mJ for the DBP baseline and 46 mJ for the Kalman Filter. Notably, despite model #3 in the DL-based approach comprising four layers and approximately 1.7k parameters, inference remains highly efficient thanks to the TensorFlow Lite runtime and ESP-NN optimizations [279]. A comparable trend is observed on the Pico 2 W and the MCXN947. Despite their different power profiles, approximately 77 mW and 431 mW, respectively, the resulting energy costs are of the same order of magnitude. This behavior can be explained by the longer inference time on the Pico 2 W, imposed by its lower clock frequency, and by the use of a dedicated hardware accelerator on the MCXN947, which enables faster execution. The ESP32 again represents a balanced compromise between execution speed and energy efficiency.

Data transmission (Tx) represents the second most energy-demanding operation after wake-up. The ESP32 and Pico 2 W exhibit similar transmission energy consumption, around 324 mJ and 347 mJ, respectively, reflecting the characteristics of their Wi-Fi modules. In contrast, the MCXN947 requires significantly less energy for transmission—almost an order of magnitude lower—thanks to its Ethernet-based communication, which is far more energy-efficient than Wi-Fi. Considering the large gap between transmission energy and the relatively low cost of prediction, PBDC-based approaches can deliver considerable energy savings when properly tuned.

Finally, additional reductions in energy consumption can be achieved by exploiting the extremely low power consumption of deep sleep modes, measured at approximately 11 mW on the Pico 2 W. Approaches such as DLDS, which extend sleep intervals by predicting multiple future samples, can therefore deliver further and potentially significant energy savings.

3.4 Experimental Results

In this section, we present the experimental validation and the obtained results of the innovative frameworks and methodologies proposed in this chapter. The analysis is organized into two primary subsections. First, we evaluate the capabilities of the proposed IoT node acting as a virtual sensor. This validation begins with the results of a real-world deployment designed to correlate continuous environmental telemetry with biological ground

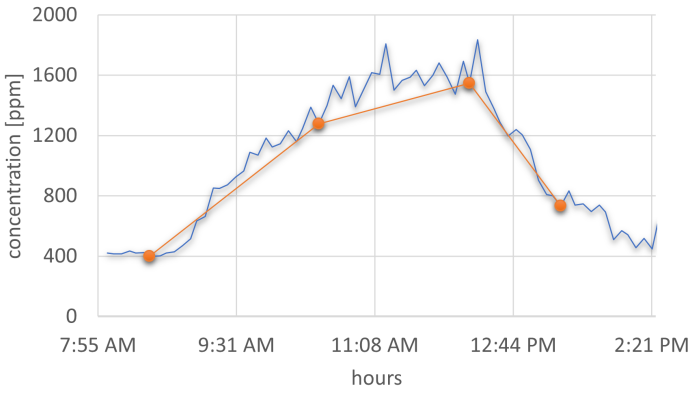
truth, quantifying the system's ability to track contamination dynamics in the air. Subsequently, we detail the computational performance of the associated ML pipeline, presenting the results of model training and assessing the specific impact of data augmentation techniques on prediction accuracy. The second section presents a comprehensive study of the Deep Learning-Based Data Collection (DLBDC) and Deep Learning Driven Sensing (DLDS) strategies, reporting the performance of the proposed forecasting models and characterizing their accuracy and efficiency compared to traditional sampling baselines.

3.4.1 Validation of the IoT Virtual Sensor

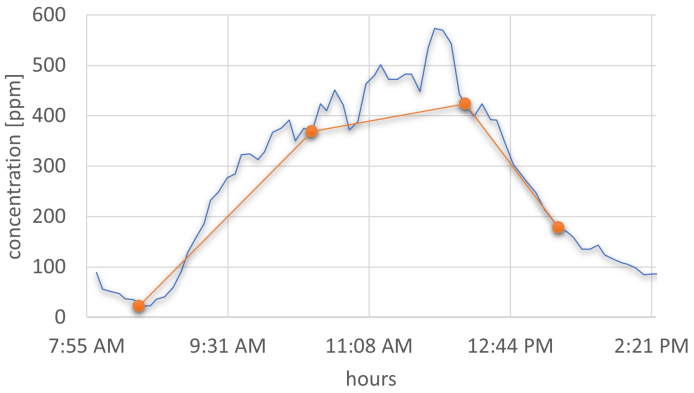
To assess the hypothesis that environmental parameters can reliably indicate biological contamination, a series of field experiments were carried out in a real-world educational environment. The resulting dataset enables examination of the relationships between human presence, air quality metrics, and microbial loads measured through conventional microbiological techniques.

Figure 3.13 presents data points collected on June 22, 2023, during a written exam attended by 14 students in a classroom of approximately 100 m^2. The plots illustrate the temporal evolution of CO_2 and VOCs represented by the continuous lines. Both CO_2 and VOC concentrations exhibited a rapid increase concurrent with student arrival (approximately 09:00 am). This growth continued, albeit with a slower gradient, until the end of the exam (around 12:00 pm). It is crucial to note that the classroom lacked a mechanical HVAC system, and only one window was partially open. Consequently, the gas concentrations tended to reach a plateau, representing the equilibrium between anthropogenic production and natural dispersion/infiltration. Superimposed on these environmental curves is an orange line connecting the discrete timestamps where the airborne bacterial load was physically measured using the Surface Air System (SAS). Samples were collected at four strategic intervals: one as a baseline at 08:30 am before student entry, one at 10:30 am during peak exam activity, one at the end of the session at 12:00 pm upon student exit, and finally, one at 01:20 pm one hour after the exam terminated. The SAS measurements confirm that the concentration of Colony Forming Units (CFU/m^3) follows a trajectory that faithfully mirrors the CO_2 and VOC profiles. This strong temporal alignment empirically validates the proposed IoT sensor as a viable starting point for indirectly estimating indoor bacterial load.

After the physical validation phase, we evaluated the ability of the DL models to predict microbial concentration levels using only environmental sensor data. The Multi-Layer Perceptron (MLP) regressor showed a strong ability to model the non-linear relationships between the input variables (CO_2, particulate matter, temperature, humidity, and TVOCs) and the biological target. Figure 3.14 illustrates the results obtained during the training, validation, and testing phases of the model, along with a cumulative plot with all three phases together. The model achieved a regression coefficient (R^2) score of 0.92, indicating a very high agreement between predicted values and ground-truth measurements. This performance remained stable under different operating conditions, including changes in occupancy levels and outdoor weather, with the prediction error consistently staying below 10%.



(a)



(b)

Figure 3.13: CO_2 (a) and VOCs (b) levels measured during a written exam on June 22, 2023. The orange line connects the points at which the bacterial load has been measured using a Surface Air System sampler (SAS).

Due to the limited size of the original labeled dataset, around 160 samples, data augmentation was essential for enhancing model robustness. The MLP was trained on datasets expanded by factors ranging from 2× to 12× using the previously described generative techniques. Among these, the variational autoencoder (VAE) produced the most consistent results, with a 10× dataset expansion yielding the most statistically stable training set. When trained solely on the original dataset, the model reached a Mean Absolute Error (MAE) of 61.7 CFU. After augmentation, the MAE decreased to 50.82 CFU, demonstrating a clear improvement in predictive performance. Practically, this level of error aligns with typical safety guidelines, where the alarm threshold for indoor biological contamination is around 300 CFU/m^3. A MAE of roughly 50 CFU/m^3 thus provides a sufficient margin to reliably detect critical conditions while minimizing false alarms.

Overall, the approach effectively integrates low-cost sensing hardware with machine

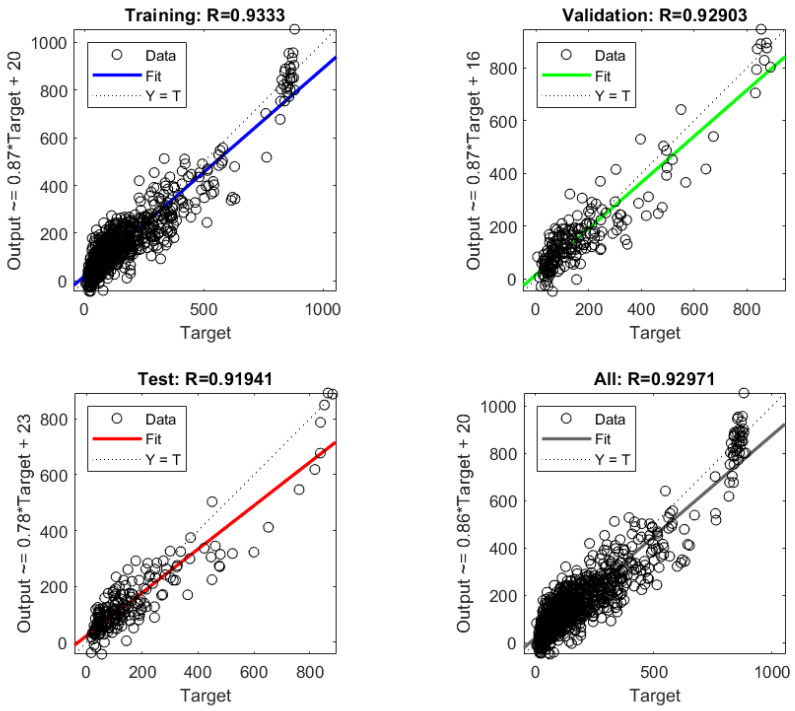


Figure 3.14: Result of the proposed MLP model during training, validation, test, and all.

learning techniques, delivering accurate, actionable predictions that support timely ventilation or sanitization decisions. However, while the MLP model demonstrates high predictive fidelity within the tested environment, its transferability to different building types remains a critical consideration. The current validation was conducted in a historical building characterized by natural ventilation. In modern facilities equipped with advanced HVAC systems or HEPA filtration, the statistical correlation between environmental proxies and airborne bacteria may weaken. Mechanical filters effectively remove biological particulate matter without necessarily altering gas concentrations. Therefore, it requires an initial calibration phase, leveraging transfer learning, when deployed in buildings with significantly different ventilation and occupancy dynamics.

Furthermore, the reliance on low-cost sensors, particularly for TVOC, temperature, and humidity monitoring, introduces challenges related to sensor drift and cross-sensitivity. These devices are known to exhibit signal degradation over time or respond to non-target organic compounds. The proposed virtual sensor addresses these hardware limitations by using distinct environmental parameters, providing a more robust estimation compared to single-proxy models. The stability of these results is empirically supported by a comprehensive data acquisition campaign conducted over an entire year, ensuring full seasonal coverage and capturing the inherent aging of low-cost hardware.

Finally, regarding large-scale deployment, the system demonstrates high local scalability due to the minimal computational overhead required for the virtual sensor's inference.

The primary consideration for massive roll-outs lies in the backend architecture, which must handle the data aggregation logic for numerous distributed nodes in real time. Because the virtual sensor transforms standard IoT nodes into biological "sentinels", its deployment on a massive scale is primarily a matter of integrating these data streams into centralized facility management platforms. This integration enables a pervasive, highly responsive, and cost-effective biological risk assessment ecosystem across entire building complexes.

3.4.2 Evaluation of the Energy-Aware Data Transmission Methodologies

After confirming the sensing capabilities and the accuracy of biological estimates, attention turns to the operational sustainability of the system. Although the IoT node is capable of reliable indoor safety monitoring, maintaining continuous high-fidelity measurements necessitates strategies for energy management.

The evaluation of PBDC-based approaches focuses on balancing measurement accuracy with power efficiency. First, the predictive performance of the forecasting models, which underpin the data reduction logic, is assessed. This is followed by an analysis of DLBDC, highlighting its effect on reducing radio transmissions, and an examination of DLDS, which extends the device's low-power sleep intervals. Collectively, these results illustrate how the system can maintain rigorous monitoring standards while optimizing energy consumption and maximizing battery life.

Performance of the forecasting models

An initial evaluation is conducted on the models' performance in a single-step forecasting scenario ($ts = 1$), which is the core mechanism for the DLBDC strategy. Table 3.3 presents a comparative analysis of three variations of our proposed Deep Learning models (Model #1, #2, and #3) against the two reference baselines: Derivative-Based Prediction (DBP) and the Kalman Filter (KF). The results, expressed in terms of Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), demonstrate that Model #3 consistently yields the highest accuracy across all four datasets. Notably, for CO_2 forecasting, Model #3 achieves a remarkable error rate of only 0.58%, corresponding to an MAE of approximately 3.6 ppm. While the performance remains robust, the error naturally increases when estimating quantities with less predictable, stochastic trends. For particulate matter ($PM_{2.5}$) and radioactivity (γ *dose rate*), the average percentage error rises to approximately 3.5% and 6.4%, respectively. Crucially, the Deep Learning approach outperforms both the DBP and Kalman Filter techniques across all datasets. The performance gap is particularly evident in the more volatile datasets (γ *dose rate*), suggesting that the DL model captures complex non-linear dependencies that linear or derivative-based estimators fail to model. Given that a higher predictive accuracy correlates directly with a higher potential for data suppression (as fewer realignments are needed), Model #3 was selected as the reference architecture for all subsequent characterization experiments.

A subsequent evaluation investigates the models' capability to predict a sequence of future values ($ts > 1$), a requirement for the DLDS. Figure 3.15 reports the degradation of model accuracy (MAPE) as the forecasting window ts increases. As anticipated, increasing the prediction horizon leads to a rise in average error. This effect is particularly noticeable

Table 3.3: Forecasting performance of the proposed Deep Learning models compared to reference approaches.

et	Model	MAE	Unit	MAPE
CO_2	Model #1	14.250	$[ppm]$	0.022
	Model #2	7.766		0.012
	Model #3	**3.597**		**0.006**
	DBP	9.518		0.014
	KF	11.251		0.017
$PM_{2.5}$	Model #1	0.233	$[\mu g/m3]$	0.063
	Model #2	0.202		0.057
	Model #3	**0.121**		**0.035**
	DBP	0.203		0.054
	KF	0.199		0.052
$Noise$	Model #1	0.076	$[mV]$	0.039
	Model #2	0.064		0.033
	Model #3	**0.037**		**0.019**
	DBP	0.064		0.033
	KF	0.066		0.034
γ *dose rate*	Model #1	33.881	$[nSv/h]$	0.122
	Model #2	34.155		0.106
	Model #3	**18.247**		**0.064**
	DBP	32.695		0.096
	KF	30.288		0.087

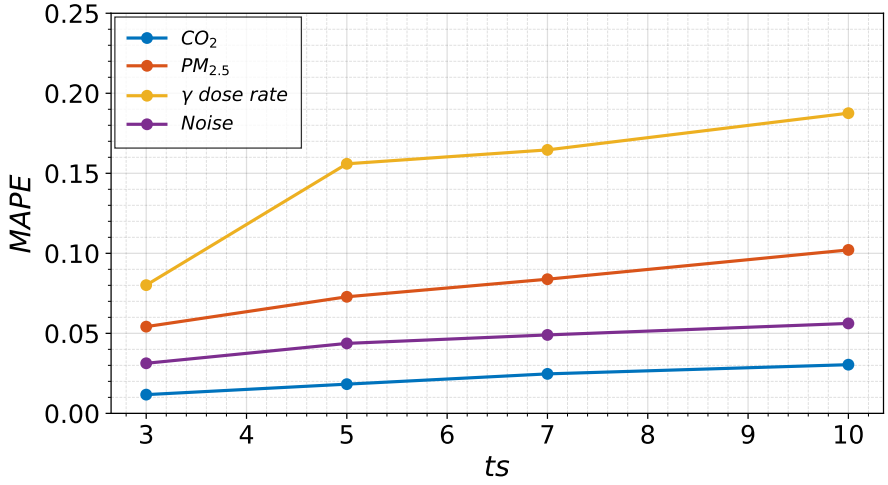


Figure 3.15: Multistep forecasting error (MAPE) when increasing the size of the prediction window (ts).

for variables that are harder to forecast, such as radioactivity, where the MAPE quickly surpasses 15%. From a design standpoint, this trade-off remains manageable. As discussed in the theoretical analysis in Section 3.2.2, the DLDS approach delivers substantial energy efficiency improvements even with small window sizes (between 3 and 4). Within this range, prediction errors stay low enough to allow reliable data reconstruction while still enabling the device to remain in sleep mode across multiple cycles.

To further optimize the DLDS strategy, we evaluate how the prediction error is distributed across the individual samples within a fixed window. Figure 3.16 plots the MAPE relative to the specific sample position ($t+1, \ldots, t+10$) for a model configured with $ts = 10$. Two clearly different error evolution patterns emerge across the analyzed signals. For CO_2, the prediction error grows approximately linearly with the forecasting horizon and exhibits a relatively small slope, indicating that the model is able to capture long-term trends effectively for this gas. This behavior suggests robust predictive performance even when the forecasting window is extended. In contrast, the remaining environmental metrics display a clearly different pattern. The error increases sharply within the first few prediction steps, up to the fourth position, and then tends to stabilize or even decrease as the horizon progresses. The slope of these curves reflects the intrinsic complexity of the signals, with more volatile quantities, such as radioactivity, exhibiting consistently higher errors.

This error distribution plays a central role in the DLDS realignment mechanism. Because the strategy assesses the accuracy of the entire multi-step forecast using only the final predicted value, denoted as $\hat{V}_{t+ts}$, the relationship between the terminal error and the errors accumulated at intermediate steps becomes critical. The observed differences between linear and non-linear error propagation indicate that the selection of the forecasting horizon ts must be carefully adapted to the characteristics of each dataset. An unsuitable horizon can trigger premature or unnecessary realignments, increasing communication overhead

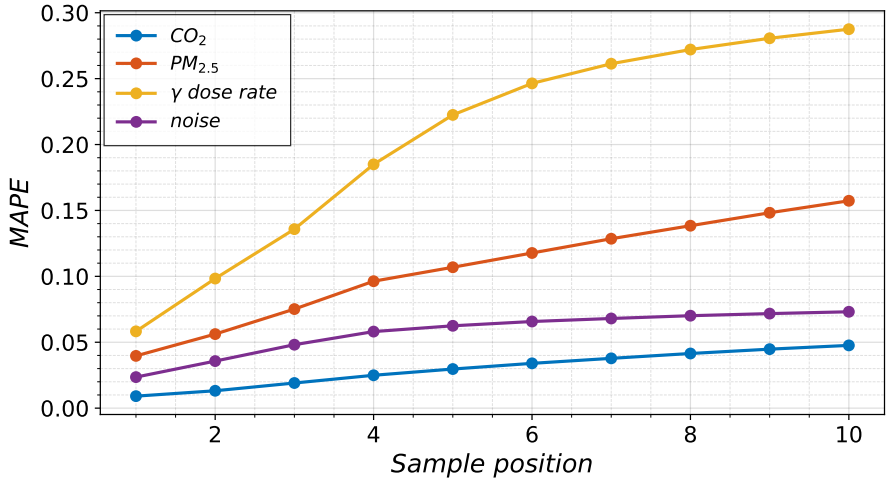


Figure 3.16: Distribution of multi-step forecasting error across sample positions for a model configured with $ts = 10$.

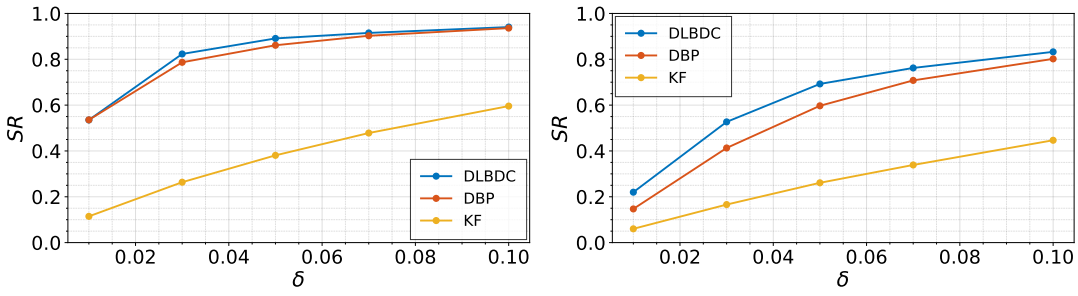
and ultimately diminishing the energy-saving benefits of the DLDS approach.

DLBDC performance

The primary objective of the PBDC mechanism is to minimize radio transmissions. We define the Suppression Rate (SR) as the ratio of the number of suppressed messages to the total number of sampling instances. Figure 3.17 presents the SR achieved by the DLBDC strategy compared to the reference systems (DBP and KF) across the selected datasets. The results are plotted against varying tolerance thresholds (ϵ), expressed as a relative percentage error (δ).

As expected, increasing the allowable relative error (δ) reduces the number of transmitted packets, since a greater proportion of predicted values fall within the specified validity bounds. As a result, the suppression rate (SR) rises monotonically with δ. The DLBDC approach consistently outperforms the two reference methods in minimizing transmissions across all three datasets (CO_2, $PM_{2.5}$, and $Noise$). This advantage is particularly pronounced in the CO_2 scenario: with a strict tolerance of 3%, DLBDC achieves a suppression rate of about 82%, compared to approximately 78% for DBP and roughly 27% for KF. These results demonstrate that permitting a small reconstruction error in CO_2 monitoring allows DLBDC to prevent the transmission of over 80% of data packets, resulting in significant energy savings.

The observed performance trends further reveal a strong dependence of the suppression rate on the underlying dynamics and predictability of the monitored signal. Signals characterized by slow temporal variations and high predictability tend to yield higher suppression rates, as the model can accurately capture their evolution over time. Indoor CO_2 concentration is a representative example, since it typically increases gradually as a result of human respiration. Conversely, more stochastic and volatile signals, such as ambient



(a) CO_2 (b) $PM_{2.5}$

(c) *Noise* (d) γ *dose rate*

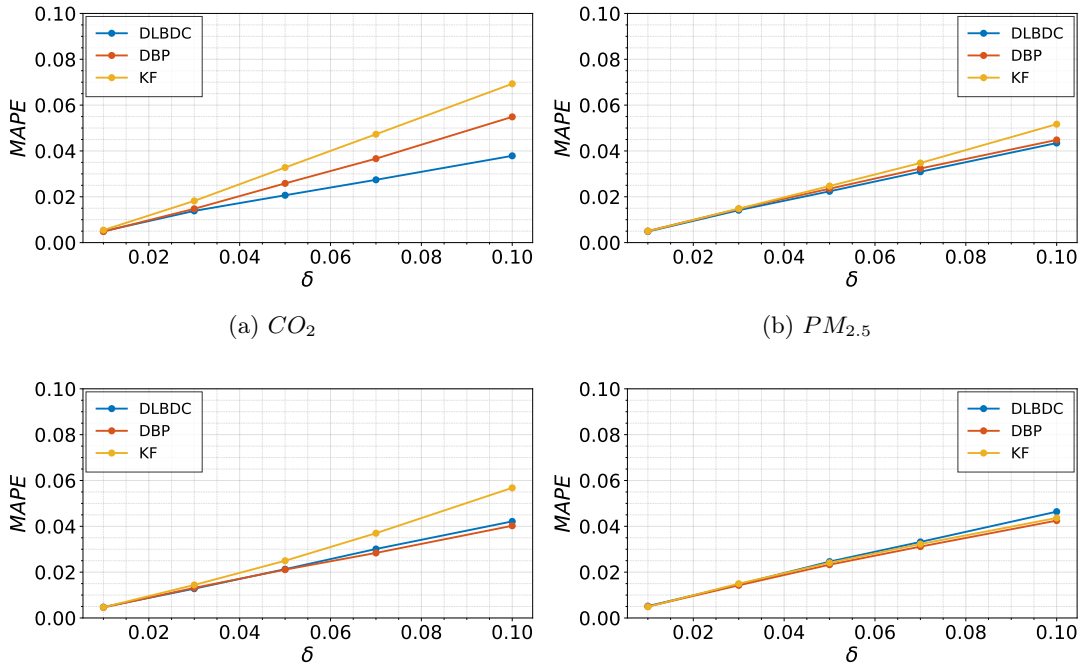
Figure 3.17: Suppression rate of the DLBDC strategy compared to reference systems (DBP, KF) as a function of the user-defined tolerance threshold δ.

Noise or γ *dose rate*, are inherently harder to predict and therefore exhibit consistently lower suppression rates. In the specific case of the γ *dose rate* dataset, the KF achieves performance comparable to that of DLBDC, while the DBP strategy shows noticeable improvements only when the relative error tolerance exceeds 3%.

Overall, for a standard tolerance level of $\delta = 0.03$, the suppression rate achieved by the proposed DLBDC strategy ranges from approximately 38% in the most challenging, high-entropy scenarios to 82% in the most stable and predictable conditions. These results highlight the versatility of DLBDC across heterogeneous sensing domains and confirm its effectiveness in significantly reducing transmission-related energy consumption in IoT systems.

While maximizing suppression is desirable, it must not compromise the utility of the collected data. Figure 3.18 illustrates the MAPE calculated on the server side (reconstructed data) as a function of the threshold δ. It is important to note that this parameter represents the theoretical upper bound on the admissible reconstruction error, since the system transmits the actual measurement whenever the prediction error exceeds this threshold. Consequently, the effectiveness of a given strategy can be assessed by how much the observed MAPE remains below the corresponding δ value.

Across all datasets, the DLBDC method consistently produces the lowest reconstruction error compared to the alternative approaches. In the CO_2 dataset, for example, DLBDC



(a) CO_2 (b) $PM_{2.5}$

(c) *Noise* (d) γ *dose rate*

Figure 3.18: Variation in the server-side reconstruction error (MAPE) of DLBDC strategy calculated for increasing values of δ.

achieves a MAPE roughly three percentage points lower than the reference methods at the same tolerance, while in the *Noise* dataset, the reduction is close to two percentage points. Because δ defines the maximum permissible error, any difference between the specified tolerance and the actual MAPE represents an additional accuracy margin. Practically, this means that relatively relaxed tolerance values can be used without degrading data quality at the cloud server. With $\delta = 0.05$, the MAPE for CO_2 remains slightly above 2%, with similar trends observed for $PM_{2.5}$, *Noise*, and γ *dose rate*. Even when the tolerance is increased up to 10%, the reconstructed signals maintain errors below 5% across all datasets.

In conclusion, the experimental results confirm that DLBDC effectively balances transmission reduction and reconstruction accuracy. The method can suppress more than 80% of data transmissions for stable and predictable signals while introducing only a marginal reconstruction error, on the order of 1.4%. Even under highly volatile conditions, DLBDC maintains a reduction of approximately 38%, with an error slightly above 1.5%, thereby demonstrating its robustness and suitability for energy-efficient IoT data acquisition.

Having established that the DLBDC approach effectively reduces packet transmissions at the software level, the subsequent step is to quantify the corresponding impact on energy consumption and battery performance. To this end, the active energy expenditure was calculated both with and without the PBDC strategies, applying Equations 3.4 and

3.2 to the device state parameters reported in Table 3.2.

Table 3.4 summarizes the total energy savings obtained with the DLBDC method, alongside the corresponding MAPE, in comparison with the reference techniques. The computation was performed over a period of 400 hours with a sampling interval of 5 minutes and using a fixed tolerance threshold of $\delta = 0.03$. The most cost-effective options for energy savings and MAPE are emphasized in bold text.

Table 3.4: Energy savings of the DLBDC, DBP, and KF methods across the three hardware platforms.

			ESP32	Pico 2W	MCXN947
Dataset	Technique	MAPE	Saving [%]	Saving [%]	Saving [%]
CO_2	**DLBDC**	**0.014**	**31.68**	**36.21**	**3.09**
	DBP	0.015	31.25	35.03	4.11
	KF	0.018	5.42	8.33	-3.07
$PM_{2.5}$	**DLBDC**	**0.014**	**19.55**	**22.06**	**1.43**
	DBP	0.015	16.10	18.36	1.28
	KF	0.015	1.49	3.12	-4.43
$Noise$	**DLBDC**	**0.013**	**28.95**	**32.24**	**3.02**
	DBP	0.013	28.06	32.13	4.17
	KF	0.014	16.88	20.15	-1.08
γ *dose rate*	DLBDC	0.015	11.02	13.23	0.34
	DBP	**0.014**	**15.65**	**17.81**	**1.02**
	KF	0.015	7.16	10.03	-2.02

The experimental results indicate a decisive advantage for the DLBDC approach when applied to wireless-enabled nodes (ESP32 and Pico 2 W). Across three datasets, this strategy consistently yielded the highest energy savings, ranging from 19% to 36% while maintaining MAPE values between 1.4% and 1.3%. In these contexts, the DL model effectively captured the signal dynamics, allowing the node to suppress a significant volume of transmissions. This performance notably exceeded that of the KF, which typically achieved only single-digit percentage savings, and consistently outperformed the DBP baseline. An exception occurred in the γ *dose rate* dataset. The inherently stochastic nature of radioactive decay limited the DL model's predictive advantage. In this case, the DBP algorithm yielded marginally higher energy savings (around 3–4%) while producing a nearly equivalent MAPE.

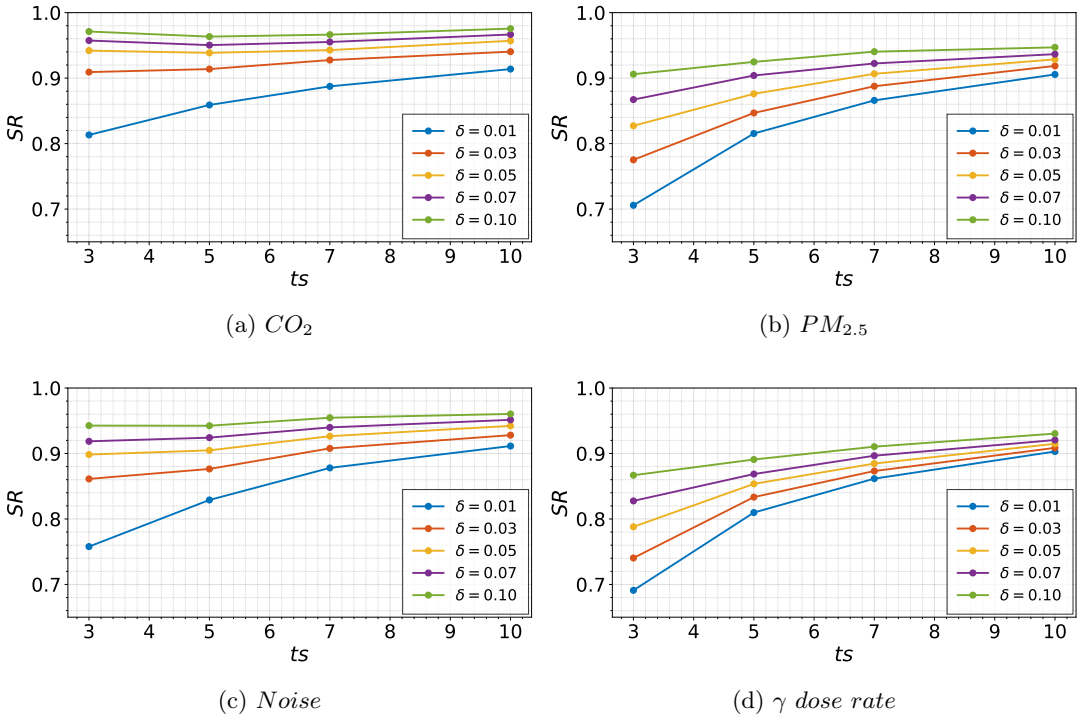
A distinct shift was observed when deploying the algorithms on the MCXN947. Unlike the wireless boards, this platform utilizes a highly efficient industrial Ethernet interface. Consequently, the DLBDC method produced limited benefits, with a maximum energy saving of only 3%. All strategies demonstrated marginal or even negative savings, meaning the device consumed more energy than the baseline. This phenomenon is attributable to the energy balance ratio $\mathcal{E}_{tx}/\mathcal{E}_{pred}$. On wireless boards, the energy cost of radio transmission is high, easily justifying the computational cost of running an inference to avoid it. However, the Ethernet interface is sufficiently efficient that the energy saved by suppressing a packet is roughly equivalent to, or sometimes less than, the energy consumed by the forecasting

model. Thus, the reduction in transmission power did not fully offset the inference overhead. In summary, the reported results demonstrate that the proposed DLBDC technique, in IoT wireless devices, can save a non-negligible portion of active energy at the cost of an accuracy reduction not exceeding 2% of the measurement.

DLDS performance

The DLDS approach enables the node to enter deep sleep for multiple consecutive time steps, potentially achieving greater energy savings than the single-step DLBDC method. Its effectiveness depends on three main parameters: the input buffer size (ws), the prediction horizon (ts), and the error tolerance (δ). Initial tests determined that $ws = 5$ provides optimal performance, and this value was kept fixed in subsequent analyses to focus on the influence of δ and ts.

The initial evaluation examined how the SR varies with different ts values and δ settings. As shown in Figure 3.19, the proportion of suppressed transmissions remains consistently high, never dropping below 70% for any of the datasets. The suppression rate



(a) CO_2 (b) $PM_{2.5}$

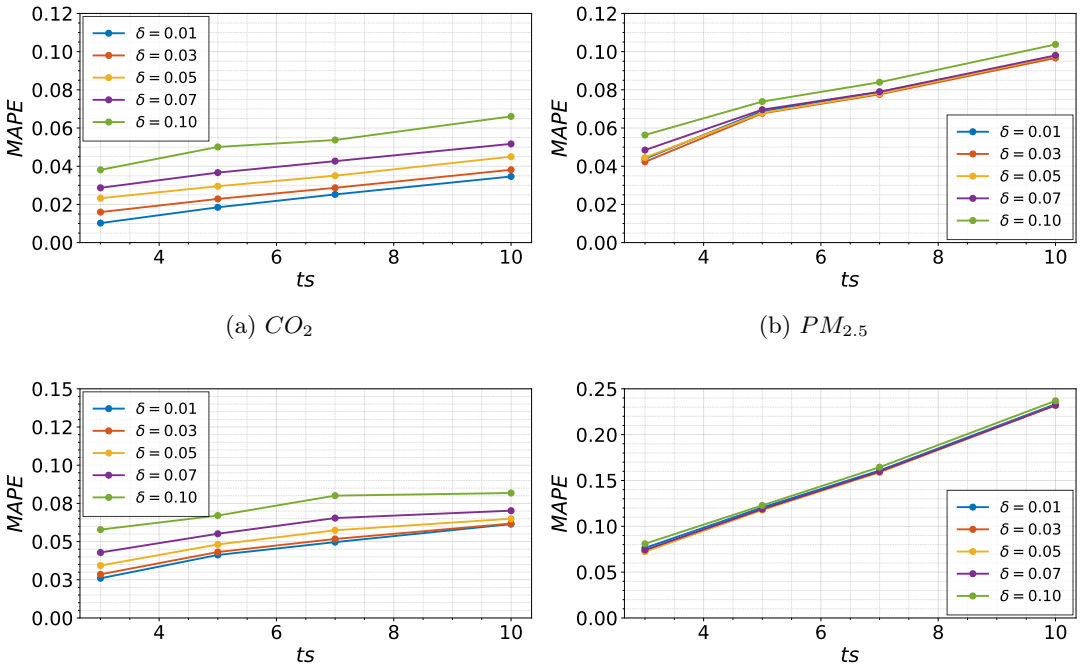
(c) *Noise* (d) γ *dose rate*

Figure 3.19: Suppression rate for the DLDS strategy when varying the value of ts combined with different configurations of δ.

(SR) grows with increasing ts, reaching its maximum at $ts = 10$. This effect is intrinsic to the algorithm, as larger ts values correspond to longer deep-sleep periods with no data transmissions. The impact of the tolerance parameter δ is more pronounced for

smaller ts values, where SR is largely dictated by the forced sleep interval. In this regime, δ determines whether the end-of-period verification passes; a failed check triggers buffer realignment, which reduces the total transmission suppression.

The trade-off for this increased suppression is visualized in Figure 3.20, which reports the server-side reconstruction error (MAPE). Across all datasets, the MAPE exhibits an



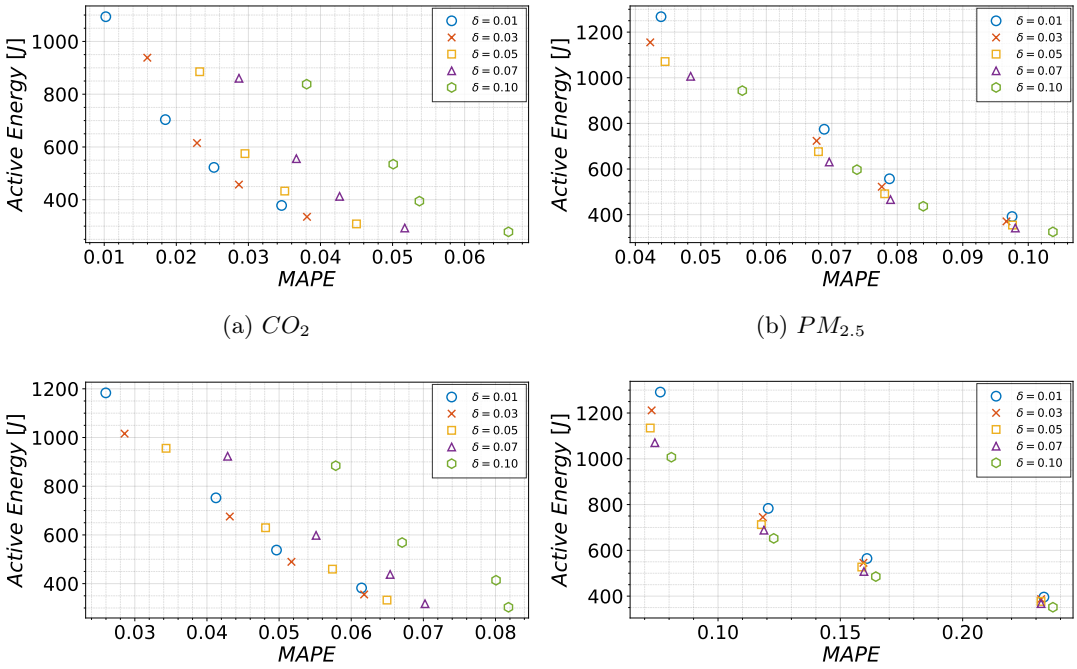
(a) CO_2 (b) $PM_{2.5}$

(c) *Noise* (d) γ *dose rate*

Figure 3.20: Reconstruction error for the DLDS strategy for increasing value of ts and δ.

almost monotonic growth as the prediction horizon extends. However, the influence of δ varies significantly by dataset. For stable quantities like CO_2, a small δ improves accuracy. Conversely, for high-entropy quantities such as γ *dose rate*, the parameter δ produces negligible effects. In these scenarios, the value predicted at position ts rarely satisfies the acceptance threshold regardless of the tolerance, consistently triggering data transmission and buffer realignment.

To identify the optimal operating point for the DLDS strategy, we analyzed the trade-off between active energy consumption and reconstruction error using Pareto charts. Figure 3.21 specifically plots the performance of the ESP32, which serves as the representative intermediate point in terms of performance among the three evaluated platforms. In this representation, points nearest the origin indicate the best trade-off between energy savings and prediction error. The analysis shows that, for all datasets except CO_2, the optimal prediction horizon is $ts = 5$, whereas for CO_2 the ideal value is $ts = 7$. An interesting pattern emerges regarding the tolerance parameter δ. For CO_2 and Noise, a tight tolerance ($\delta = 0.01$) produces the lowest MAPE. In contrast, for $PM_{2.5}$ and γ *dose rate*, the optimal



(a) CO_2 (b) $PM_{2.5}$

(c) *Noise* (d) γ *dose rate*

Figure 3.21: Pareto charts plotting active energy versus the server-side reconstructed error.

settings involve looser tolerances of $\delta = 0.07$ and 0.10, respectively. This behavior suggests that imposing strict error limits on highly variable data triggers frequent buffer realignments, which can inadvertently destabilize predictions and reduce overall performance.

Finally, Table 3.5 summarizes the best achievable trade-offs for each dataset, assuming a standard constraint where the cloud server accepts a maximum measurement error of 3%. A critical distinction from the DLBDC strategy is that none of the platforms exhibit

Table 3.5: Reconstructed error together with the energy saving for the DLDS strategy when $\delta = 0.03$

			ESP32		Pico 2W		MCXN947	
Dataset	MAPE	Steps [#]	En. [J]	Saving [%]	En. [J]	Saving [%]	En. [J]	Saving [%]
CO_2	0.029	7	457	88.23	425	**89.11**	568	78.34
$PM_{2.5}$	0.030	2	1,155	57.28	1,570	**58.66**	1,993	22.19
Noise	0.028	3	1,015	74.05	937	**75.42**	799	69.14
γ dose rate	*0.053*	*2*	*1,793*	*54.33*	*1,695*	*55.10*	*2,012*	*21.07*

negative savings. Thanks to the nature of the DLDS skipping sensor acquisition and anticipating entry into deep sleep always reduces power consumption, regardless of the energy cost. Consequently, even the industrial MCXN947 achieves positive savings, although it consistently remains the most conservative of the three boards. In the case of the CO_2, for example, the system can utilize a large horizon ($ts = 7$), resulting in massive energy savings of approximately 89%. The same Noise dataset meets the constraint at $ts = 3$,

providing 75% savings. Conversely, for the $PM_{2.5}$, to satisfy the 3% error limit, the horizon must be restricted to $ts = 2$, yielding savings of roughly 58%. Unfortunately, for the γ *dose rate*, even at the minimum horizon ($ts = 2$), the stochastic nature of the data prevents the system from meeting the 3% error constraint. The actual measured error was approximately 5%, with energy savings of 54%.

In summary, the experiments show that the DLDS strategy is capable of cutting active energy consumption by up to 89% through the use of extended deep-sleep periods. At the same time, the results highlight that its applicability depends on the nature of the monitored quantity. While highly effective for environmental gases, the approach is less suitable for Poisson-distributed measurements such as γ *dose rate*, where unpredictability limits the feasibility of long sleep intervals.

Limitations

While the experimental results highlight the significant potential of the two proposed PBDC frameworks, it is essential to acknowledge specific limitations inherent to the design and valid boundaries for its application. Furthermore, these constraints naturally suggest several directions for future investigation and refinement. A first limitation concerns the need for domain-specific modeling. Each physical quantity to be monitored requires a dedicated forecasting model that must be trained and deployed accordingly. Although this approach ensures that the predictive mechanism is well aligned with the intrinsic dynamics of the signal (i.e., the slow rise of CO_2 or the rapid fluctuation of $Noise$), it introduces a non-negligible setup cost during the preliminary system design. The effectiveness of the proposed strategies is also strongly dependent on the temporal characteristics of the sensed data. As confirmed by the experimental results, the highest gains are achieved when the monitored signals exhibit predictable trends. In contrast, signals dominated by stochastic behavior, such as the γ *dose rate*, are inherently harder to forecast accurately. In these cases, the increased frequency of data transmissions or buffer realignments reduces the achievable energy savings. Another critical factor is the underlying communication technology. For battery-powered wireless platforms, such as the ESP32 and Pico 2 W, both DLBDC and DLDS provide substantial energy benefits by significantly reducing the number of transmissions. Conversely, for devices relying on wired communication, such as industrial IoT nodes equipped with Ethernet interfaces, the advantages become marginal or negligible. In these scenarios, the energy cost of data transmission is already low, and the computational overhead associated with neural network inference is not sufficiently compensated by packet suppression. Finally, unlike traditional IoT architectures where the server simply logs incoming data, the proposed framework requires a stateful backend. The cloud server must maintain a synchronized history buffer for each active sensor node to support the prediction mechanism. As a result, the memory requirements grow linearly with the number of deployed devices. While this overhead remains manageable in typical small- to medium-scale deployments, it becomes a critical consideration for large-scale IoT systems involving thousands of nodes.

3.5 Summary

The methodology described in this chapter addresses the gap between theoretical indoor safety standards and the practical constraints of continuous monitoring systems. By combining low-cost IoT nodes with distributed AI at the edge, the approach extends traditional environmental sensing toward real-time biological risk assessment. This architectural paradigm is based on two key elements: the definition of a virtual sensor capable of inferring airborne bacterial loads from conventional environmental measurements, and the adoption of predictive data management strategies aimed at mitigating the trade-off between high-frequency sampling requirements and long-term energy sustainability.

The experimental results validate the efficacy of this approach on both fronts. Regarding biological monitoring, the field tests confirmed a strong correlation between human occupancy proxies, specifically CO_2 and Volatile Organic Compounds (VOCs), and the proliferation of biological contaminants. The Multi-Layer Perceptron (MLP) model demonstrated exceptional precision in translating these proxies into bacterial load estimates (CFU/m^3), achieving a coefficient of determination (R^2) of 0.92. Crucially, the integration of generative data augmentation techniques, such as Variational Autoencoders (VAE), significantly improved the model's robustness, reducing the Mean Absolute Error (MAE) to approximately 50 CFU. This level of accuracy proves that a software-defined virtual sensor can effectively replace expensive biological instrumentation, providing reliable triggers for sanitization interventions without the need for manual sampling. However, it is crucial to acknowledge the inherent limitations of proxy-based environmental monitoring. The virtual sensor approach relies heavily on the statistical correlation between environmental proxies and the biological target. In environments equipped with advanced air purification systems, such as High Efficiency Particulate Air (HEPA) or activated carbon filters, this correlation may weaken. Because these filters can physically remove airborne bacteria without proportionally reducing gas concentrations, the model might require specific recalibration in such settings to avoid false negatives or misestimations of the actual biological risk. Provided that reliable physicochemical proxies are identified for new targets, this virtual sensing framework can be readily exported to other critical domains, such as urban air quality monitoring for pollution hotspots or smart agriculture applications.

Parallel to the sensing capabilities, the evaluation of the Deep Learning-Based Data Collection (DLBDC) and Deep Learning-Driven Sensing (DLDS) strategies demonstrated that high-resolution monitoring is compatible with long-term autonomy. By shifting the computational load from the cloud to the Edge, the system successfully utilized predictive models to suppress redundant radio transmissions and extend sleep intervals. The DLBDC strategy allowed for the suppression of up to 82% of data packets while maintaining a reconstruction error below 1.5%, effectively minimizing the most energy-intensive operation of the IoT node. Furthermore, the DLDS approach, by forecasting future trends and enabling prolonged deep sleep states, yielded energy savings of up to 89% on standard MCUs like the ESP32 and Raspberry Pi Pico 2 W. These findings collectively demonstrate that by embedding intelligence directly into the device, it is possible to achieve a sustainable, high-fidelity monitoring infrastructure capable of supporting the rigorous demands of modern hygiene management.

Chapter 4

Vision-Based People Monitoring

While environmental and biological sensing provide direct data on air quality, the presence and movement of people represent the primary drivers of contamination and the main variables in indoor safety management. Since human occupancy is strictly correlated with the increase in CO_2 levels, the suspension of airborne pathogens, and the wear of surface hygiene, real-time tracking of human presence becomes a key element for intelligent sanitization. However, current monitoring solutions often struggle to distinguish between simple presence and semantic occupancy patterns, or they fail to address the critical trade-off between granular data collection and individual privacy. These limitations frequently result in either invasive surveillance or a complete lack of actionable data for hygiene management.

This chapter examines the use of Computer Vision as a virtual sensing mechanism to extract spatiotemporal information about human presence, moving beyond static cleaning schedules toward a dynamic, risk-based approach. The primary research contribution is a privacy-by-design occupancy pipeline. By performing all image processing directly on edge hardware, the system ensures that sensitive visual data never leaves the sensor node, effectively addressing privacy concerns while providing real-time metrics on crowd dynamics and flow. To ensure the sustainability of this high-computational task, the framework incorporates a Dynamic Inference Power Manager (DIPM) system that intelligently modulates the execution of detection and tracking algorithms based on historical occupancy trends. The proposed approach is assessed through an extensive comparison of classical and lightweight tracking methods, considering both their accuracy and computational cost. Overall, the results show that vision-based virtual sensing can deliver reliable occupancy information with limited energy requirements, making it suitable for supporting automated and effective hygiene management strategies.

4.1 Background

The paradigm of facility management has undergone a profound transformation in recent years, shifting progressively from static, schedule-based routines to dynamic, demand-driven strategies. This transition is particularly evident in the context of smart buildings, where cleaning, sanitization, and energy distribution processes are increasingly required to adapt to actual space usage rather than predefined timetables. In this specific domain, the

ability to accurately monitor human presence represents a fundamental enabling factor. Decisions regarding when to activate cleaning procedures, or deploy sanitization resources must be based on reliable information about how spaces are occupied throughout the day [280]. This capability, commonly referred to as occupancy intelligence, aims to provide real-time and fine-grained data describing space utilization patterns. Unlike simple binary presence detection, true occupancy intelligence requires a deeper understanding of the environment, encompassing not only whether a space is occupied, but also how many individuals are present and how they move within it [281]. Such information is critical for enabling adaptive strategies that balance service quality with energy efficiency, activating sensing and cleaning systems only when necessary and reducing unnecessary power consumption.

Before the widespread adoption of digital imaging, the monitoring of human presence relied heavily on non-visual electronic sensors. While these technologies offered a cost-effective entry point for early smart building applications, they present distinct limitations when challenged with the complex semantic requirements of modern facility management. The most prevalent solution in this domain has historically been the Passive Infrared (PIR) sensor. These devices operate by detecting changes in infrared radiation (heat signatures) emitted by moving bodies within their immediate surroundings [282]. Despite their low cost and simplicity, PIR sensors suffer from a critical inability to detect passive occupancy. As noted by various researchers, if an occupant remains stationary (i.e., sitting at a desk) or leaves the room while leaving warm belongings behind, the sensor lacks the sensitivity to detect this change [283]. This limitation frequently leads to erroneous classifications of vacancy, resulting in premature system shutdowns that disrupt user comfort. Alternative approaches have attempted to leverage Radio-Frequency (RF) signals to bridge this gap. Technologies utilizing Wi-Fi and Bluetooth track the Received Signal Strength Indicator (RSSI) from mobile devices to infer presence [284]. While these methods are useful for general analytics, they are constrained by low spatial resolution, with positioning errors typically ranging from 3 to 10 meters. Furthermore, their reliability is dependent upon occupants actively carrying specific smart devices, rendering them unsuitable for high-security or passive monitoring scenarios [280]. More recently, Time of Flight (ToF) sensors have evolved as a non-invasive alternative. By measuring the time taken for light pulses to travel to an object and return, ToF sensors create distance maps without capturing identifiable personal data, thus preserving privacy [283]. However, while they excel at geometric measurements, they often lack the semantic understanding necessary to distinguish between humans and other dynamic objects in complex scenes.

To overcome the intrinsic limitations of traditional sensing technologies, Computer Vision (CV) has emerged as a powerful alternative for advanced occupancy intelligence. Unlike PIR or ToF sensors, which infer presence from indirect proxies such as heat signatures or distance measurements, CV enables the direct algorithmic interpretation of visual data. This capability allows systems to reliably discriminate between human and non-human entities and to detect passive or stationary occupants, effectively bridging the gap between low-level data acquisition and high-level semantic understanding of the scene.

CV as a formal scientific discipline traces its origins to the 1960s [285]. However, a pivotal theoretical turning point occurred in the late 1970s and early 1980s with the work

of David Marr. Marr proposed a computational framework suggesting that the human visual system functions by recovering 3D structures from 2D images projected onto the retina. This insight steered the research community toward techniques based on *shape from X*, where X represents visual cues such as motion, texture, or shading aimed at mathematically reconstructing the depth lost during image projection [285]. For several decades following these early developments, the field was characterized by a reliance on handcrafted features. In this period, researchers manually designed algorithms, such as SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients), to mathematically define which visual elements were relevant for recognition, creating a rigid but interpretable framework [286].

Although neural networks enjoyed popularity in the 1980s and 90s via the back-propagation algorithm, they fell into disuse in the early 2000s due to insufficient training data and computational power [287]. The renaissance of the field occurred in 2012 with AlexNet, which demonstrated the overwhelming superiority of Convolutional Neural Networks (CNNs) in classifying massive image datasets. By 2014, computer vision entered a new era. Modern deep learning models automatically learn hierarchical feature representations directly from data, surpassing the need for manual feature engineering. [286, 287].

4.1.1 Object Detectors

At its core, an object detector is a computational model designed to answer the two fundamental questions of computer vision: "What objects are present, and where are they located?" [286]. While image classification assigns a single label to an entire image, object detection involves both classification and localization, drawing bounding boxes around regions of interest. The evolution of these detectors mirrors the broader history of the field. The first real-time performance was achieved in 2001 by the Viola-Jones detector, which utilized a sliding window approach. This was followed in 2005 by the Histogram of Oriented Gradients (HOG), a method motivated specifically by pedestrian detection, and in 2008 by the Deformable Part-based Model (DPM), which treated objects as collections of flexible parts (i.e., limbs and torso) rather than rigid templates [286]. The landscape shifted permanently in 2015 with the introduction of R-CNN (Regions with CNN features), and YOLO (You Only Look Once) [288]. By 2025, modern detectors like YOLOv9 and RT-DETR continue to lead the field in accuracy and efficiency [289].

The innovations presented in this part of the thesis focus on the deployment of person tracking algorithms on resource-constrained devices, where computational power, memory, and energy availability are inherently limited. In this context, conventional server-grade object detectors are typically unsuitable due to their high computational and resource demands. Therefore, investigate the suitability of lightweight architectures designed specifically for mobile and edge inference. In particular, we considered three popular object detectors: (i) SSD MobileNet, (ii) TrafficCamNet, and (iii) ResNet.

SSD MobileNet v2 was introduced by Google in 2018 as part of the second generation of the MobileNet family. Specifically designed to address the constraints of mobile and embedded computer vision applications [290]. The model was created to provide a reasonable trade-off between detection accuracy and computational efficiency, making it

particularly suitable for real-time inference on resource-constrained devices. At its core, SSD MobileNet v2 adopts the MobileNet v2 architecture as a lightweight feature extractor. The network begins with a standard fully convolutional layer composed of 32 filters, followed by 17 residual bottleneck modules. Each module is built around the concept of *inverted residuals* and *depth-wise separable convolutions*, which significantly reduce the number of parameters and floating-point operations compared to conventional CNN layers. Structurally, these modules consist of a 1×1 convolutional layer for channel expansion, a 3×3 depth-wise convolution for spatial feature extraction, and a subsequent 1×1 projection layer, with ReLU6 used as the activation function to enhance numerical stability on low-precision hardware. To enable object detection, the backbone is coupled with the Single Shot Detector (SSD) framework [291]. In this configuration, the network is naturally divided into two functional components: the backbone, responsible for extracting multi-scale visual features, and the detection head, implemented through the SSD layers, which perform object classification and bounding box regression in a single forward pass. The integration of Feature Pyramid Network (FPN) concepts allows the detector to operate on feature maps at different resolutions, improving robustness to scale variations while maintaining low inference latency. Thanks to this modular design and its emphasis on efficiency, SSD MobileNet v2 has become a widely adopted baseline for edge AI applications, particularly in scenarios where real-time person detection and tracking must be achieved under strict computational and energy constraints.

TrafficCamNet is a deep learning model developed within the NVIDIA TAO framework and optimized for running on edge platforms. This detector is built upon the DetectNet v2 architecture and employs ResNet-18 as its backbone for feature extraction [292], providing a balanced compromise between detection accuracy and computational efficiency. TrafficCamNet is specifically trained for surveillance and monitoring scenarios characterized by high or elevated camera viewpoints, a common configuration in smart buildings, public facilities, and large indoor environments. The model is capable of detecting objects belonging to four predefined classes: (i) cars, (ii) persons, (iii) two-wheelers, and (iv) road signs. Although originally designed for traffic analysis, its robustness in identifying people from overhead or oblique perspectives makes it suitable for occupancy monitoring applications in indoor and semi-indoor contexts. From an architectural standpoint, TrafficCamNet follows the GridBox object detection paradigm. Input frames, typically resized to a resolution of 960×544, are divided into a regular grid, and the network performs bounding box regression by predicting localization parameters and confidence scores for each grid cell. These raw predictions are subsequently refined through post-processing steps based on clustering techniques such as Non-Maximum Suppression (NMS) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). This post-processing stage is essential to suppress redundant detections and to produce the final bounding box coordinates and class labels with improved spatial consistency. Thanks to its integration within the NVIDIA ecosystem and its optimization for Jetson devices, TrafficCamNet represents a practical and efficient solution for real-time person detection in edge-based occupancy intelligence systems.

The **ResNet** architecture was introduced in 2015 as a major advancement in deep convolutional neural networks, and it has since become a foundational model in computer

vision due to its strong performance and efficient training behavior [293]. ResNet addresses one of the key limitations of very deep CNNs, namely, the vanishing gradient problem, which often degrades accuracy as the number of layers increases. The core innovation of ResNet lies in the introduction of residual blocks, which restructure the learning process through *skip connections*. Rather than forcing each stack of layers to learn a direct mapping from input to output, a residual block learns only the difference (the residual) between the input and the desired output. Within the ResNet family, ResNet-10 represents a lightweight variant composed of 10 layers, making it particularly suitable for deployment on resource-constrained and edge devices [294]. Compared to deeper configurations such as ResNet-50 or ResNet-101, ResNet-10 offers a significantly reduced computational footprint while preserving a favorable accuracy-to-complexity ratio. Despite its compact structure, it is capable of delivering competitive performance in visual recognition tasks, especially in scenarios where computational efficiency, low latency, and energy consumption are critical. Although ResNet architectures are frequently employed as backbones within more complex detection pipelines, the ResNet model itself remains a fundamental building block for modern computer vision systems, providing a robust and efficient foundation for edge-oriented inference.

4.1.2 Object Trackers

While object detection focuses on the spatial localization of targets within individual frames, it is inherently memoryless, as each image is processed independently and without temporal context. In contrast, the analysis of indoor environments, such as tracking people moving through corridors, entering or exiting rooms, or forming occupancy patterns over time, requires the integration of temporal information. This need is addressed by Object Tracking, which extends frame-level detection by enforcing temporal coherence across successive observations. An object tracker is a computational algorithm designed to follow the temporal evolution of one or more targets across consecutive frames. Its main goal is to preserve object identity over time, thereby enabling the reconstruction of motion trajectories. Formally, at time t, the tracker receives the set of bounding boxes B_t generated by the detector and associates them with the set of active trajectories T_{t-1}. This association process typically involves: (i) assigning unique identifiers (IDs) to newly detected objects, (ii) updating the state of existing tracks, and (iii) terminating tracks that are no longer observed.

Multi-Object Tracking (MOT) has its roots in the 1960s, when it was first developed in the context of aerospace systems using Kalman filtering techniques to track radar targets [295]. In modern computer vision, MOT is predominantly addressed through the tracking-by-detection paradigm, which decomposes the problem into two consecutive stages. First, an object detector is applied independently to each frame to localize all candidate targets. Second, a tracking module performs data association, linking current detections to existing trajectories across time. By maintaining consistent identities for objects as they move through the scene, this approach enables the reconstruction of trajectories that capture complex motion dynamics and occupancy flows, providing valuable insights into human movement patterns within indoor environments.

A fundamental distinction in the tracking literature concerns the classification of algorithms as *Online* or *Offline*. Offline tracking algorithms process an entire video sequence as a batch and are therefore able to estimate the state of an object at frame t using information from both past frames $(t-1, t-2, \dots)$ and future frames $(t+1, t+2, \dots)$. While this look-ahead capability often results in higher accuracy and improved robustness to occlusions, it inherently precludes real-time operation. Online tracking algorithms, in contrast, infer the object state at frame t using only the information available up to that instant. All decisions must be made immediately, with minimal latency, to produce actionable outputs in real time.

Our research focuses on monitoring human movement and reacting to changes in occupancy in real-time. For this reason, online tracking represents the only viable solution, as occupancy information must be updated continuously to enable the system to respond promptly to evolving conditions within the environment. Furthermore, the deployment of the proposed solutions on edge devices introduces strict constraints in terms of computational resources, memory availability, and energy consumption, effectively excluding the use of overly complex tracking pipelines. To satisfy these real-time and hardware constraints, the computational complexity of the tracking algorithm must be carefully considered.

The **Intersection Over Union (IOU)** is a lightweight tracking algorithm proposed by Bochinski et al. in 2017 [89]. The fundamental assumption of this approach is that, given a sufficiently high frame rate, the position of an object changes minimally between frame $t-1$ and frame t. Therefore, the overlap between the detection in the current frame and the prediction from the previous frame serves as a robust proxy for identity association. The Intersection over Union measure is mathematically defined as:

$$IOU(a, b) = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \tag{4.1}$$

Where a represents the bounding box of a detection in the current frame, and b represents the bounding box of the last known position of an active track.

Figure 4.1 shows a visual representation of the IOU algorithm. Consider a situation in which an active track from the previous frame, indicated as $BBOX_0$, must be associated with candidate detections in the current frame. The detector outputs two bounding boxes, $BBOX_1'$ and $BBOX_1''$. For each candidate, the tracker evaluates the intersection areas, denoted as I' and I'', and computes the corresponding IOU values. In this case, the overlap I' produces a noticeably higher IOU than I'', leading the algorithm to associate $BBOX_1'$ with the existing track $BBOX_0$ and update its estimated position accordingly.

This association mechanism follows a set of rules that regulate the creation, update, and termination of trajectories:

1. Each active trajectory is compared against all detections in the current frame, and the detection with the highest IOU is selected, provided that this value exceeds a predefined threshold σ_{IOU}.

2. Any detection in the current frame that cannot be associated with an existing trajectory is treated as a new object, and a corresponding trajectory is initialized.

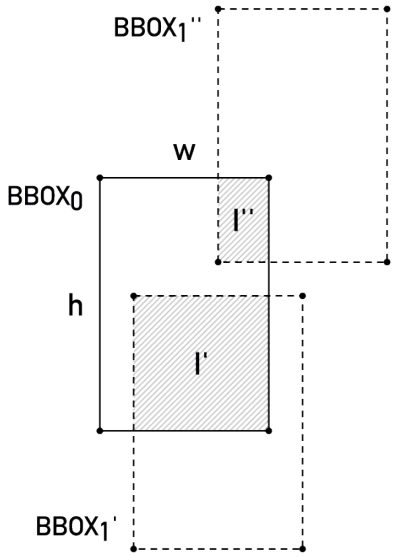


Figure 4.1: Visual representation of the IOU association mechanism.

3. Any trajectory from the previous frame that cannot be associated with any detection in the current frame is considered lost or occluded and is therefore terminated.

4. To reduce the impact of spurious detections and short-lived false positives, a temporal validation step is applied. A trajectory is considered valid for occupancy estimation only if its duration exceeds a minimum time threshold t_{min}. Trajectories shorter than this threshold are discarded.

Another important parameter used inside the IOU tracker is the σ_l. Its purpose is to filter the detections produced by the object detector based on a confidence score to remove every detection that has low confidence, since it most likely represents an error.

The IOU tracker is a relatively simple algorithm that requires minimal computational resources, making it a popular choice for basic tracking tasks and real-time applications on resource-constrained devices. However, its limitations emerge in complex scenarios characterized by a high number of objects, frequent interactions, partial occlusions, and significant variations in object scale.

The **Simple Online and Realtime Tracking (SORT)** is a multi-object tracking framework that prioritizes computational speed for online applications [296]. The algorithm distinguishes itself by decoupling object detection from the state estimation, employing standard control theory techniques to achieve high accuracy with minimal resource overhead. The SORT algorithm operates by assigning detections to existing targets through a combination of motion prediction and data association. Unlike the IOU tracker, which assumes an object's position is static between frames, SORT employs a Kalman Filter to model the motion dynamics of each target. The algorithm assumes a linear constant velocity model. For every tracked object, the filter predicts its future location in the next frame

based on its current position and velocity. This predictive capability allows the system to bridge gaps where the object detector fails or when a target is briefly occluded, as the filter provides an estimated location even in the absence of a visual confirmation. Once the Kalman filter has predicted the new positions of all active tracks, these predictions must be matched with the actual bounding boxes provided by the detector in the current frame. SORT utilizes the Hungarian Algorithm to solve this assignment problem. The algorithm computes an optimal matching that minimizes the error between the predicted states and the detected objects, ensuring that the most likely detection is linked to the correct identity.

New tracks are not immediately confirmed upon first detection. Instead, they enter a probationary phase during which they must be successfully associated with detections for a minimum number of consecutive frames, denoted as min_hits. This temporal validation mechanism ensures that only objects exhibiting consistent presence are promoted to active tracks, effectively suppressing sporadic false positives produced by the detector. To prevent uncontrolled memory growth and the persistence of stale or ghost trajectories, tracks are explicitly terminated when they are not updated for a predefined number of frames. If a track fails to receive an associated detection for a duration exceeding the threshold T_{lost}, it is marked as lost and removed from the tracking pool. By combining a constant-velocity motion model with efficient data association based on spatial proximity, SORT is able to handle short-term occlusions and moderate pose or scale variations substantially better than simple IOU-based trackers. These characteristics make it well-suited for the dynamic indoor scenarios addressed by the proposed smart building monitoring framework.

The **NvDCF** is a proprietary multi-object tracking algorithm developed by NVIDIA and built on top of the Discriminative Correlation Filters (DCF) [297]. This tracker represents a significant departure from standard tracking-by-detection paradigms by incorporating visual feature learning directly into the tracking loop. The core mechanism involves learning a target-specific correlation filter for each detected object. This filter encodes the visual appearance of the target and is used to re-identify it in subsequent frames. For every active target, the tracker defines a search region around its expected position in the next frame. Applying the learned correlation filter to this region, the algorithm computes a correlation response map. The peak of this map indicates the highest probability of the target's new location. This allows the system to maintain tracking continuity even when the object detector fails to return a bounding box, providing robustness against partial or full occlusions.

To enhance visual tracking robustness, NvDCF integrates a Kalman filter–based state estimator. As in SORT, this module predicts the target's kinematic state, providing trajectory smoothing and constraining the search area for the correlation filter. A notable strength of NvDCF lies in its tight optimization for NVIDIA platforms, as it is natively included in the DeepStream SDK. The tracker's behavior can be adjusted through the *feature size* parameter, which controls the spatial resolution of the features used during correlation. This parameter accepts integer values between 1 and 5. Smaller values favor higher computational efficiency at the expense of tracking precision, whereas larger values improve accuracy and robustness while increasing computational demand.

4.1.3 MOT Challenge

To objectively assess and compare the performance of the tracking algorithms, standard benchmarking protocols are required. The MOT Challenge (Multiple Object Tracking Benchmark) serves as the international standard in this domain. It provides a unified framework specifically designed to rigorously test tracking algorithms against common pitfalls, such as occlusion, motion blur, and crowded environments [298]. Among the various benchmarks provided by the framework, we focus on MOT17, one of the most widely adopted in the literature. This challenge dataset consists of 14 video sequences (divided equally into 7 for training and 7 for testing), capturing diverse scenarios ranging from pedestrian streets to shopping malls. Crucially, these sequences are filmed using both stationary and moving cameras, testing the tracker's ability to compensate for motion. To isolate tracking performance from detection quality, MOT17 provides three sets of public detections for each sequence, obtained via distinct object detectors: (i) DPM, (ii) Faster-RCNN, and (iii) SDP.

The participants of the challenge are required to process these sequences and generate a standard Comma-Separated Value (CSV) output file. This file lists the frame number, the assigned target ID, and the bounding box coordinates for every tracked object. To quantify performance, the generated trajectories are evaluated using the TrackEval tool [299]. The metrics computed by the software can be divided into three distinct families, each highlighting a different aspect of tracking performance: (i) HOTA metrics, (ii) CLEARMOT metrics, and (iii) Identity metrics.

HOTA metrics are designed to provide a balanced assessment by unifying detection, association, and localization errors into a single score [300]. This addresses the limitations of older metrics that often overemphasize detection quality over association capability. These metrics include Higher Order Tracking Accuracy (HOTA), Detection Accuracy (DetA), Association Accuracy (AssA), Detection Recall (DetRe), Detection Precision (DetPr), Association Recall (AssRe), Association Precision (AssPr), and Localization Accuracy (LocA). CLEAR MOT metrics focus heavily on the precision of object location and the consistency of ID assignment [301]. Those metrics include Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Multi-Object Detection Accuracy (MODA), CLEAR Detection Recall (CLR_Re), CLEAR Detection Precision (CLR_Pr), CLEAR Detection True Positives (CLR_TP), CLEAR Detection False Negative (CLR_FN), CLEAR Detection False Positives (CLR_FP), Identity switch (IDSW), Mostly tracked trajectories (MT), Partially tracked trajectories (PT), Mostly lost trajectories (ML), and Fragmentation (Frag). Identity metrics [302] view the problem through the lens of identification, emphasizing the system's ability to maintain correct identities over long durations rather than just frame-to-frame matching. Those metrics include Identification Recall (IDR), Identification Precision (IDP), Identification True Positives (IDTP), Identification False Negatives (IDFN), Identification False Positives (IDFP), and Detected Objects (Dets).

4.2 Research Contribution

This section describes the methodological framework developed to improve both efficiency and reliability of computer vision systems deployed on edge platforms for indoor monitoring. The focus is on the design of a vision-based virtual sensor and on the definition of a complete processing pipeline, spanning from raw image acquisition to the extraction of high-level information about the observed scene.

A key component of the framework is the tight coupling between object detection and tracking pipelines and a dedicated metric extraction layer. This layer translates raw visual outputs into interpretable spatio-temporal indicators, such as occupancy counts and motion patterns, which are then exploited to inform adaptive sanitization and ventilation policies. To ensure sustained operation on resource-limited platforms, the architecture incorporates a Dynamic Inference Power Manager (DIPM). This module modulates the computational effort in response to current monitoring demands, enabling significant reductions in energy consumption while maintaining the reliability and continuity of the derived metrics.

4.2.1 The Vision-Based Virtual Sensor

The proposed virtual sensor is composed of five distinct but interconnected components: the *camera sensor* is the hardware interface responsible for acquiring raw video frames, the *object detection network* is a deep learning model that identifies objects of interest within the frame, the *object tracking algorithm* is a lightweight algorithm that associates detections across time to maintain object identities, the *metrics extractor* is a logic module that processes tracking data to derive high-level analytics, and the *Dynamic Inference Power Manager (DIPM)* is a control unit that optimizes energy consumption by instructing the camera to skip frames when the environment is static. The data flow within this architecture is depicted in Figure 4.2. The blue arrows represent the flow of visual data and extracted features through the processing chain. The orange arrows indicate the state information utilized by the DIPM to assess scene activity. Finally, the green arrow represents the feedback control signal generated by the DIPM, which actively regulates the frame rate of the camera to minimize power usage during idle periods.

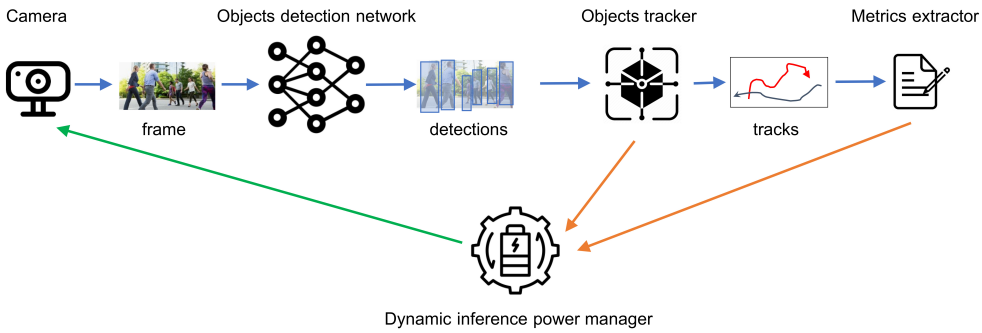


Figure 4.2: Schematic representation of the pipeline of the visual-based virtual sensor.

Metric Extractor

The metrics extractor serves as the analytical core of the virtual sensor. It acts as an abstraction layer, converting the raw bounding boxes and IDs provided by the CV modules into actionable numerical data. The algorithm accepts three primary inputs from the upstream Object Detector and Tracker: (i) the type of object detected (i.e., person), (ii) the unique identifier assigned to the object to distinguish it from others in the scene, and (iii) the bounding box coordinates and dimensions of the object $(x, y, width, height)$. By aggregating this information over temporal windows, the extractor derives the following key metrics:

- The total number of unique objects detected during a specific interval.
- The average velocity of tracked objects moving through the scene.
- The average total distance covered by the objects during their lifespan within the frame.

A fundamental limitation of 2D monocular vision systems is perspective distortion. Due to projective geometry, objects closer to the camera appear larger and exhibit greater apparent motion than identical objects located farther away, even when their true physical speed is the same. If left uncorrected, this effect introduces a systematic bias in motion and distance estimates, overemphasizing activity in the foreground. To mitigate this issue without relying on explicit camera calibration or scene geometry reconstruction, the system adopts a diagonal normalization strategy. Specifically, both speed and displacement measurements are normalized by the length of the diagonal of the corresponding bounding box. By expressing pixel-level motion relative to the object's own apparent size, the resulting metrics become scale-invariant. As a consequence, the movement of individuals near the camera is weighted consistently with that of individuals located in the background, enabling a more uniform and perspective-robust estimation of motion dynamics across the entire field of view.

The virtual sensor uses the computed metrics for various high-level processing tasks, including: estimating the number of people inside a room, monitoring objects crossing a specific passage or virtual line, calculating the duration a person spends in a specific area of interest, and computing the vectors of movement within the environment. The extracted metrics are post-processed and sent periodically to a cloud server for further analysis and storage.

Privacy is a core design requirement of this system. To protect users and ensure confidentiality, no raw visual information, such as images or video streams, is stored locally or transmitted outside the device. The data sent to the cloud consists exclusively of anonymous, aggregate numerical values (i.e., "3 people detected", "Average activity level: High"). This approach ensures strict compliance with privacy standards while maintaining the utility of the sensor for building management.

Dynamic Inference Power Manager

In conventional setups, vision sensors capture frames at fixed rates (commonly 30–60 fps) and process each frame independently, regardless of the activity in the scene. While this ensures smooth temporal coverage, it also incurs significant computational overhead. In many indoor environments, such as offices, classrooms, or corridors, human motion is often slow, sporadic, or absent for long periods. Processing every frame under these conditions yields largely redundant information, wasting energy and straining the limited resources of edge devices.

The Dynamic Inference Power Manager (DIPM) is a control algorithm specifically designed to exploit this temporal redundancy. Its core objective is to adapt the effective processing frame rate to the observed motion in the scene by estimating how many future frames (N) can be safely skipped without compromising tracking continuity or object identity. When objects move slowly or remain stationary, the DIPM increases the number of skipped frames to reduce computational load; conversely, when rapid motion is detected, the system processes frames more frequently to preserve tracking accuracy. By dynamically modulating inference frequency based on object motion, the DIPM enables significant energy savings while maintaining reliable real-time tracking, making it particularly well-suited for continuous monitoring applications on resource-constrained edge devices.

The challenge of predicting the number of frames to skip is non-trivial. It requires guaranteeing that when the system wakes up after N frames, the object tracker will still be able to associate the new position of the object with its previous track. Since the system primarily uses an IOU-based tracker, the association relies on the overlap between the previous bounding box and the current one. Therefore, the DIPM must ensure that:

$$IOU(a_{f_t}, a_{f_{t+N}}) >= \sigma_{IOU} \tag{4.2}$$

where a_{f_t}, and $a_{f_{t+N}}$ are the a-th object at the current frame, and its predicted position after N frames, respectively.

Consider an object at frame f_t enclosed by a rectangular bounding box $A'B'C'D'$ with width w' and height h'. Let (S_x, S_y) be the estimated speed of the object in pixels/frame. Under the assumption that the object does not change its dimensions over time, its predicted position after N frames can be approximated by a translated bounding box $A''B''C''D''$. As illustrated in Figure 4.3, the intersection area I between the current and predicted bounding boxes depends on the displacement over N frames and can be written as:

$$I(N) = (w' - N \cdot S_x)(h' - N \cdot S_y) \tag{4.3}$$

Accordingly, the union area U is given by:

$$U(N) = 2w'h' - I(N) \tag{4.4}$$

By substituting these expressions into the standard IOU formulation, the overlap between

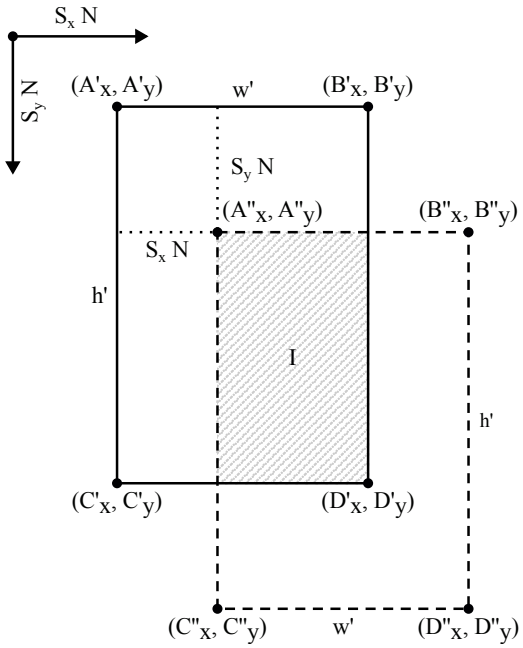


Figure 4.3: Diagram showing the bounding box at the current frame and the predicted bounding box after N frames.

the object at frame f_t and at frame f_{t+N} becomes:

$$IOU(a_{f_t}, a_{f_{t+N}}) = \frac{(w' - N \cdot S_x)(h' - N \cdot S_y)}{2w'h' - (w' - N \cdot S_x)(h' - N \cdot S_y)} \tag{4.5}$$

The objective of the DIPM is therefore to determine, for each tracked object, the maximum integer value N such that the IOU constraint remains satisfied. Using Equation 4.2 and the expressions derived above, this condition can be equivalently written as:

$$\frac{(w' - NS_x)(h' - NS_y)}{2w'h' - (w' - NS_x)(h' - NS_y)} >= \sigma_{IOU} \tag{4.6}$$

Notice that N is computed using the object speed estimated from the last two detections, under the assumption that this speed remains constant over future frames. In practice, however, object motion can vary significantly. If the object accelerates shortly after the estimation, N is overestimated, increasing the risk of losing the association with the existing track. Conversely, if the object decelerates, N is underestimated, resulting in a missed opportunity to skip additional frames.

The implementation of this logic is detailed in Algorithm 3. For each frame, the algorithm receives as input the set of current *detections* associated with their IDs assigned by the tracker, the maximum allowable number of frames to skip n_{max}, the IOU threshold σ_{IOU}, and a speed caution factor α. The number of frames to skip ($f2s$) is initially set to

Algorithm 3 Pseudo-code of the Dynamic Inference Power Manager.

Require: $detections$ ▷ List of detections from the tracker
Require: n_{max} ▷ Max numbers of frames to skip
Require: σ_{IOU} ▷ IOU threshold
Require: α ▷ Speed caution factor
Ensure: $f2s$ ▷ Number of frames to skip

$f2s \leftarrow n_{max}$
for each $det \in detections$ **do**
  $S_x, S_y \leftarrow computeSpeed(det)$
  $n \leftarrow 1$
  repeat
    $I \leftarrow (det.w - n * S_x * \alpha) * (det.h - n * S_y * \alpha)$
    $A \leftarrow det.w * det.h$
    $IOU \leftarrow I/(2 * A - I)$
    $n \leftarrow n + 1$
  until $(IOU > \sigma_{IOU}$ **and** $n <= n_{max})$
  $n \leftarrow n - 1$
  if $(n < f2s)$ **then**
    $f2s \leftarrow n$
  end if
end for

its maximum value n_{max}. The algorithm then iterates over all detected objects, estimating the object speed from its displacement between the current and the previous frame. This speed estimate is scaled by the factor α and used to predict the object's future position and the corresponding IOU value for each subsequent frame, starting from the next one. The prediction proceeds until the IOU falls below the threshold σ_{IOU} or the maximum skip limit n_{max} is reached. During this process, $f2s$ is continuously updated by retaining the minimum value obtained across all objects. In scenes with multiple targets, the fastest-moving object determines the maximum number of frames that can be safely skipped, since it is the first to violate the IOU constraint. The parameter α acts as a cautionary factor on the speed estimation. Increasing its value effectively amplifies the estimated speed, leading to a smaller predicted intersection area I and thus a more conservative skip decision. Conversely, lower values of α enable more aggressive frame skipping, improving energy savings at the cost of a higher risk of tracking failures under sudden accelerations.

4.3 Experimental Setup

4.3.1 Software Framework

The implementation, training, and deployment of the neural networks and algorithms were conducted using a suite of standard open-source frameworks. The software stack utilized throughout the experiments is outlined below, providing an overview of the tools integrated into the development pipeline.

The core machine learning workflow was built upon the **TensorFlow** ecosystem [267].

For the purposes of this experimental analysis, we utilized the standard Keras API for the training and validation of the networks. For deployment on edge devices, we employed LiteRT (formerly known as TensorFlow Lite) to convert and optimize the models for mobile inference. Specifically for the ESP32 platform, we utilized LiteRT for Microcontrollers (formerly TFLite Micro), a specialized variant designed to execute highly compressed models in bare-metal environments with extreme resource constraints. Additionally, deployment on the Google Coral required the use of the **Edge TPU Compiler** to translate the quantized models into executables compatible with the TPU architecture [303].

To manage the interaction with the Google Coral hardware, we utilized **PyCoral**. This is an open-source library developed by Google specifically to streamline the deployment of LiteRT models on Edge TPU devices [304]. While primarily designed for Python, PyCoral offers bindings for C++, Java, and Node.js, allowing for versatile integration into various application environments. In our system, PyCoral served as the bridge to the hardware accelerator, managing the loading of the compiled models and handling the data transfer to and from the TPU. The library is backed by a vibrant developer community and provides a repository of pre-trained models for common computer vision tasks, which significantly simplifies the prototyping phase.

On the NVIDIA Jetson platform, high-throughput video analysis was implemented using the **DeepStream SDK**. Built atop the GStreamer multimedia framework [305], DeepStream enables full GPU utilization for video inference and supports constructing complex pipelines via configuration files or custom plugins. A central feature is its integrated multi-object tracker plugin, offering multiple tracking logics. In this study, *DeepStream 6.0.1* was used to benchmark the performance of its built-in IOU and NvDCF trackers.

As an alternative to the complexity of DeepStream, we also employed **Jetson Inference**. This is a streamlined framework created by NVIDIA to facilitate the rapid deployment of AI applications on Jetson devices [306]. Jetson Inference provides a comprehensive set of C++ and Python APIs, along with a library of pre-trained deep learning models optimized for tasks such as image classification, object detection, and semantic segmentation. It is particularly well-suited for robotics and IoT applications where low latency is critical. However, compared to DeepStream, it offers more limited tracking capabilities. Natively, the framework only provides a basic IOU tracker implementation, lacking the advanced correlation filters found in the DeepStream SDK.

4.3.2 Embedded Devices

Three distinct hardware platforms were selected to evaluate the algorithms across a broad range of edge computing capabilities. Together, these devices span the entire spectrum of edge computing, ranging from high-performance embedded systems to specialized AI accelerators and ultra-constrained microcontrollers, thereby reflecting the diverse architectural and computational paradigms found in real-world deployments.

For high-performance edge AI experiments, the NVIDIA **Jetson Orin Nano** [307] was used. This embedded platform features a quad-core Arm Cortex-A57 CPU and 8 GB of high-bandwidth RAM, providing sufficient resources for demanding AI workloads. NVMe

storage support ensures that I/O operations do not limit performance. Connectivity options include USB, HDMI, Ethernet, and GPIO interfaces, enabling flexible integration with external devices. At its core, the Orin Nano incorporates an NVIDIA GPU based on the Maxwell architecture, with 1024 CUDA cores and 32 Tensor cores, allowing massive parallelism and efficient execution of complex deep learning models. Development is supported through the NVIDIA JetPack SDK, which provides a full software stack including a Linux OS, CUDA-X libraries, and TensorRT for optimized inference.

To explore the benefits of dedicated hardware acceleration, we adopted the **Google Coral AI** platform [308]. Its defining component is the on-board Tensor Processing Unit (TPU), a specialized co-processor capable of delivering up to 4 trillion operations per second (TOPS) using 8-bit fixed-point arithmetic. This design allows neural networks to be executed with extremely low latency and power consumption. The system is built around a quad-core Arm Cortex-A53 system-on-chip (SoC), paired with 1 GB of LPDDR4 RAM and 8 GB of eMMC flash storage. Exploiting the Edge TPU requires models to be fully quantized to 8-bit integers and compiled specifically for the accelerator. The Coral ecosystem provides a complete development framework based on TensorFlow Lite and Coral APIs, simplifying deployment and interaction with the hardware. This tight hardware–software integration makes Coral well-suited for edge applications such as real-time computer vision, machine learning, and other inference-intensive tasks, including more advanced scenarios like pipelined execution across multiple TPUs.

The third platform targets the extreme low-power end of the edge. For this purpose, we used the **ESP32** SoC. A detailed description of its architecture and specifications is provided in Chapter 3.3.5. Within this experimental setup, the ESP32 is used to evaluate the feasibility of deploying the lightest configurations of the proposed neural networks on hardware characterized by very limited SRAM and computational resources.

4.4 Experimental Results

4.4.1 Trackers Characterization

The evaluation of the vision-based virtual sensor concentrates on its capability to deliver accurate and reliable occupancy information while preserving the level of energy efficiency required for continuous operation on edge devices. Because dynamic sanitization strategies rely on a precise and stable representation of human presence, the experimental analysis addresses three tightly connected aspects of system performance. First, the robustness of the tracking stage is examined by comparing different tracking algorithms and verifying their ability to follow individuals consistently across a range of indoor conditions. This is followed by an assessment of how effectively the system converts raw trajectories and movement patterns into accurate occupancy and flow metrics, ensuring that the extracted information faithfully represents real human behavior. Finally, the efficiency of the Dynamic Inference Power Manager is analyzed, with particular attention to its ability to lower energy consumption by adjusting the computational load in response to real-time scene dynamics. Taken together, these results show that the proposed system can meet the strict monitoring requirements of health-sensitive environments while significantly extending the

operational lifetime of the edge hardware.

Optimization of the IOU Tracker

The performance of the IOU tracker is heavily dependent on its configuration parameters. To quantify this sensitivity and identify the optimal setup, we conducted two sets of experiments on the MOT17 benchmark: a deterministic analysis that varied one parameter at a time, and a stochastic Monte Carlo analysis.

In the deterministic experiments, we varied a single input parameter linearly while holding the others constant. For each step, the entire MOT17 benchmark was executed to observe the impact on key metrics. Figure 4.4 illustrates the response of six representative performance metrics to changes in the IOU threshold (σ_{iou}), the detection confidence threshold (σ_l), and the minimum track length (t_{min}). Specifically, as the IOU thresh-

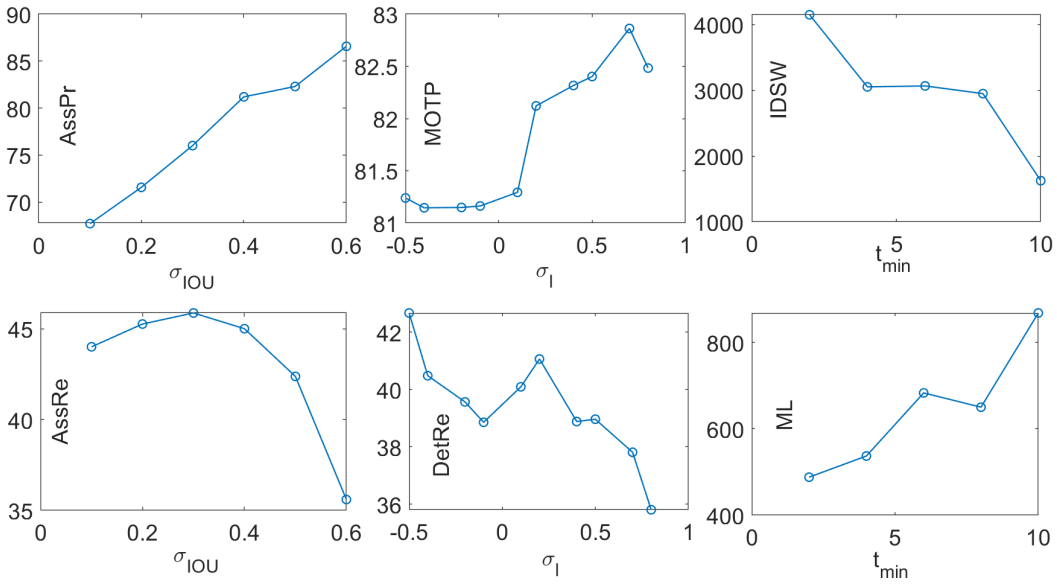


Figure 4.4: Performance metrics trends when varying the three IOU parameters.

old σ_{iou} is increased, the association precision (AssPr) improves significantly, rising from approximately 67% to 85%. This indicates that higher overlap requirements successfully filter out ambiguous associations. However, this strictness penalizes the system's ability to maintain associations under challenging conditions, causing the association recall (AssRe) to degrade from 45% to 35%. A parallel behavior is observed for the detection confidence threshold σ_l. Higher values improve the Multiple Object Tracking Precision (MOTP) by discarding uncertain bounding boxes, but inevitably lower the detection recall (DetRe), as valid, but lower-confidence detections are ignored. The impact of the minimum track length parameter, t_{min}, reveals a different but equally critical trade-off regarding track stability versus completeness. As shown in the right-hand graphs, increasing its value yields a substantial reduction in the identity switch (IDSW) counter, which drops from

over 4,000 to approximately 2,000. This suggests that filtering out short-lived tracks effectively removes the noise that contributes to spurious identity switches. However, this stability comes at the expense of tracking completeness as the mostly lost (ML) metric increases continuously. This confirms that while a higher t_{min} produces cleaner, more stable trajectories, it incorrectly discards valid short trajectories, marking those objects as lost.

To overcome the limitations of testing single parameters in isolation, we conducted a global sensitivity analysis using a Monte Carlo approach. We performed 200 experiments by pseudo-randomly sampling the design space of the three independent variables: the IOU threshold (σ_{iou}), the detection confidence threshold (σ_l), and the minimum track length (t_{min}). Table 4.1 summarizes the interactions between these configuration parameters and the MOT17 performance metrics. The values represent the Pearson correlation coefficients, providing a quantitative measure of how each input parameter influences the tracking performance. Each metric was also labeled with an arrow identifying the desired trend sign (i.e., an arrow pointing up indicates that the metric should be maximized and minimized, respectively).

The most significant correlations (with absolute value greater than or equal to 0.5) are highlighted in boldface and discussed in the following:

- σ_{iou} shows a strong positive correlation with association precision (AssPr). Increasing the parameter value forces the tracker to be more conservative, accepting only high-overlap associations. While this minimizes false matches and significantly reduces track fragmentation (Frag), it negatively impacts association recall (AssRe), as valid associations with lower overlap, such as fast-moving objects, are rejected. However, since increasing σ_{iou} improves nearly all maximization metrics (up-arrow) and reduces minimization metrics (down-arrow) with the sole exception of recall, a relatively high value is generally preferable for track purity.

- The parameter σ_l functions as a strict confidence threshold on the detector's outputs. Setting high values filters out a larger portion of detections, which strongly decreases recall-based metrics such as detection recall (DetRe), ID recall (IDR), and ID true positives (IDTP), because many valid detections are ignored. At the same time, this filtering improves precision-oriented metrics, including detection precision (DetPr) and Multi-Object Tracking Precision (MOTP), and reduces ID false positives (IDFP). These effects indicate that σ_l is the most sensitive parameter to tune, with its optimal value closely tied to the reliability of the chosen detector. Moreover, the observed strong correlation with precision motivates a potential enhancement: employing a *weighted IOU* that penalizes associations between bounding boxes with very different confidence scores, further improving tracking accuracy.

- Analysis on the t_{min} suggests that this parameter should be kept low. While increasing its value strongly reduces identity switches (IDSW), this benefit is largely artificial. The correlation with mostly lost (ML) indicates that the reduction in switches is achieved simply by discarding shorter trajectories entirely rather than stabilizing them. Therefore, the improvement in IDSW comes at the unacceptable cost of losing valid tracks.

Table 4.1: Results of the sensitivity analysis expressed by the correlation coefficients between independent variables (IOU parameters – columns) and dependent results (MOT17 metrics - rows).

Metric	Parameters		
	σ_{IOU} [0.1 - 0.6]	σ_l [-0.5 - 0.8]	t_{min} [2 - 12]
Higher order tracking accuracy - HOTA ↑	0.058	-0.12	0.092
Detection Accuracy - DetA ↑	0.337	-0.427	-0.155
Association Accuracy - AssA ↑	-0.215	0.165	0.233
Detection Recall - DetRe ↑	0.025	**-0.838**	-0.218
Detection Precision - DetPr ↑	**0.510**	**0.776**	0.124
Association Recall - AssRe ↑	**-0.694**	-0.006	0.194
Association Precision - AssPr ↑	**0.929**	0.324	0.019
Localization Accuracy - LocA ↑	**0.550**	**0.700**	0.092
Multiple Object Tracking Accuracy - MOTA ↑	0.492	0.316	0.049
Multiple Object Tracking Precision - MOTP ↑	0.431	**0.757**	0.119
Multi-Object Detection Accuracy - MODA ↑	**0.509**	0.288	-0.009
CLEAR Detection Recall - CLR_Re ↑	0.025	**-0.804**	-0.221
CLEAR Detection Precision - CLR_Pr ↑	0.496	**0.757**	0.111
CLEAR Detection True Positives - CLR_TP ↑	0.025	**-0.804**	-0.221
CLEAR Detection False Negative - CLR_FN ↓	-0.025	**0.804**	0.221
CLEAR Detection False Positives - CLR_FP ↓	-0.458	**-0.799**	-0.137
Identity switch - IDSW ↓	0.215	-0.339	**-0.722**
Mostly tracked trajectories - MT ↑	-0.389	**-0.524**	-0.396
Partially tracked trajectories - PT ↓	0.004	**-0.716**	**-0.574**
Mostly lost trajectories - ML ↓	0.161	**0.689**	**0.542**
Fragmentation - Frag ↓	**-0.533**	**-0.682**	-0.466
Identification Recall - IDR ↑	-0.242	**-0.623**	-0.008
Identification Precision - IDP ↑	0.214	**0.766**	0.254
Identification True Positives - IDTP ↑	-0.242	**-0.623**	-0.008
Identification False Negatives - IDFN ↓	0.242	**0.623**	0.008
Identification False Positives - IDFP ↓	-0.276	**-0.878**	-0.244
Detected Objects - Dets ↑	-0.313	**-0.940**	-0.200

The Monte Carlo simulations reveal inherent trade-offs among parameters: no single configuration simultaneously maximizes all evaluation metrics. For example, raising the IOU threshold σ_{iou} tends to increase precision by enforcing stricter matching but reduces recall by discarding valid associations. To manage these competing objectives, a multi-objective optimization framework was adopted. Figure 4.5 illustrates the resulting Pareto graph, plotting HOTA (Higher Order Tracking Accuracy) against MOTA (Multiple Object Tracking Accuracy). Each point represents a different parameter set, and the Pareto

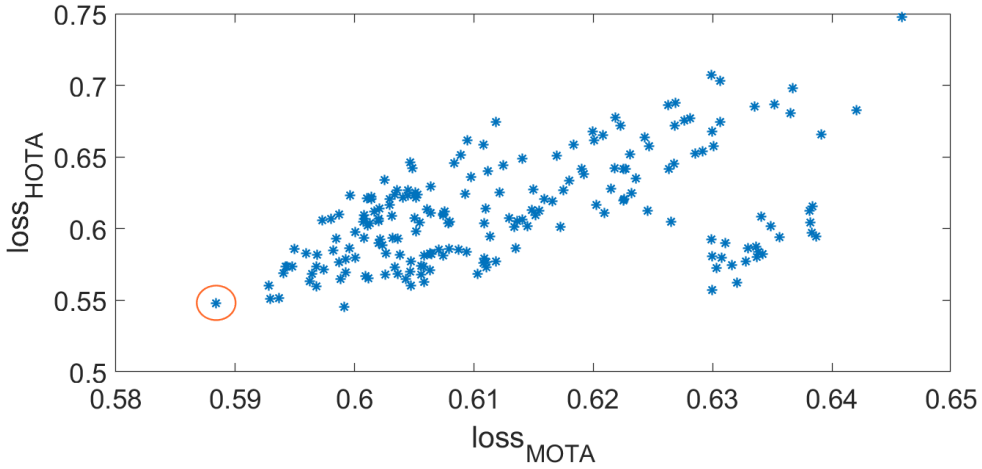


Figure 4.5: Pareto graph showing the best tradeoff point (highlighted in red), which minimizes both HOTA and MOTA losses.

frontier identifies configurations where any improvement in one metric necessarily entails a compromise in the other. This visualization aids in selecting an optimal balance between detection/tracking accuracy and association robustness for the targeted occupancy monitoring scenario. By analyzing the loss on both axes, we identified the optimal configuration as the point closest to the origin (highlighted in red), representing the best global trade-off. This configuration was obtained with the following parameters: $\sigma_{iou} = 0.4$, $\sigma_l = -0.2$, and $t_{min} = 4$.

Characterization on Edge Devices

Having established the algorithmic properties and optimal parameters of the IOU tracker, we now shift our focus to the physical constraints of the edge devices. We evaluate the energy footprint of different tracking algorithms when executed on the target hardware. The goal is to quantify the energy cost of adding a tracking stage to the detection pipeline and to understand how this cost scales with algorithmic complexity. For these tests, we utilized a standard benchmark video. The scene contains high traffic density, resulting in approximately 2,600 total object detections generated by the SSD MobileNet v2 model. The energy overhead introduced by each algorithm, using default parameters, is defined as the differential energy consumption between the full pipeline (Detection + Tracking) and the standalone detector.

Figure 4.6 compares the overhead of three tracking families: the IOU tracker, the SORT, and the NvDCF. The experimental evaluation highlights a significant disparity

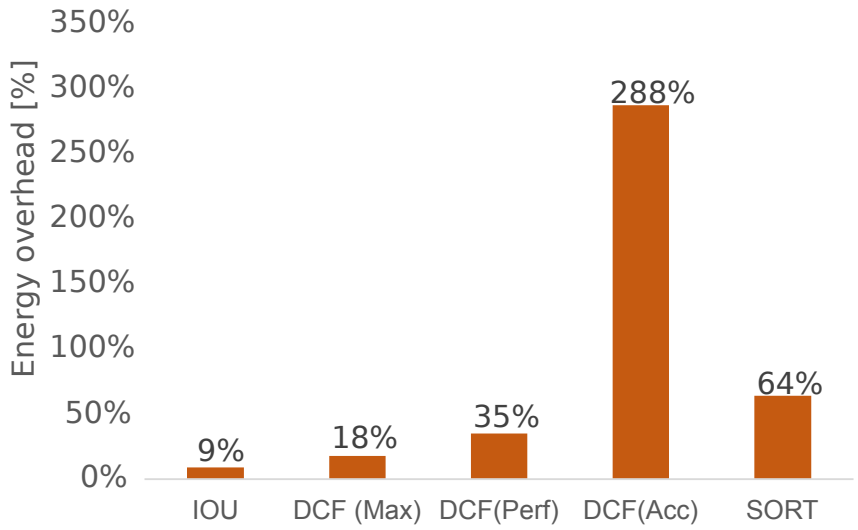


Figure 4.6: Energy overhead of the tracking algorithms with respect to the sole object detection.

in computational efficiency among the tested tracking algorithms. The IOU tracker is extremely lightweight, adding only about 9% overhead relative to the baseline detection energy, making it highly suitable for low-power edge deployment. In contrast, the NvDCF tracker was tested under three configurations reflecting different trade-offs between speed and accuracy. Energy consumption scales with algorithmic complexity. The NvDCF(Acc) configuration incurs a 288% increase over the baseline and fails to achieve real-time performance on the target edge device, dropping the frame rate to approximately 10 FPS. The SORT algorithm introduces a moderate overhead of around 64%, situating it between IOU and NvDCF in terms of energy cost. Overall, these results identify IOU as the most suitable option for energy-constrained edge applications, where minimizing power consumption is prioritized over robust performance in highly complex tracking scenarios.

We further investigated whether the energy consumption of these algorithms could be optimized by tuning their internal parameters. Figure 4.7 illustrates the energy consumption (mJ/frame) when running the SSD MobileNet v2 coupled with the IOU tracker on the Coral AI and Jetson Nano platforms. In particular, the figure reports the sensitivity to the confidence threshold (σ_l) and to the minimum track length (t_{min}), respectively. The flat trends in both graphs indicate that the computational cost of the IOU algorithm is effectively constant, as variations in these parameters do not produce noticeable changes in energy consumption. From a hardware perspective, the Coral AI demonstrates higher efficiency for this workload, consuming approximately 134 mJ/frame compared to 163 mJ/frame on the Jetson Nano. The dashed lines represent energy consumption when running only the baseline detector, confirming that the energy overhead of the IOU remains minimal regardless of the configuration.

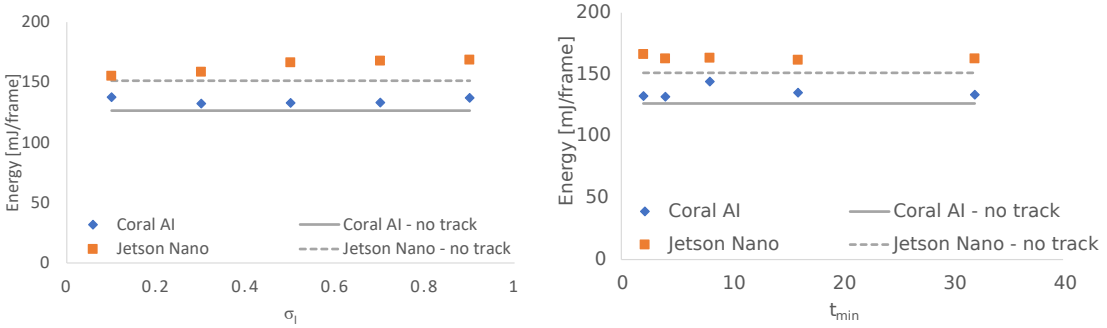


Figure 4.7: Energy consumption of SSD MobileNet v2 + IOU on both platforms when varying the σ_l (a), and t_{min} (b) parameters.

To complete the analysis, we extended the sensitivity characterization to the more sophisticated SORT and NvDCF tracking algorithms. Due to platform compatibility, SORT was evaluated on the Coral AI device, while the proprietary NvDCF was tested on the Jetson Nano. For each algorithm, one representative parameter was selected in order to observe its impact on power consumption. For SORT, the *min hits* parameter dictates the minimum number of consecutive detections required to validate a track. While for NvDCF, the *feature size* defines the resolution of the visual patches extracted for the correlation filter. Figure 4.8 illustrates the energy overhead trends for both algorithms. The energy

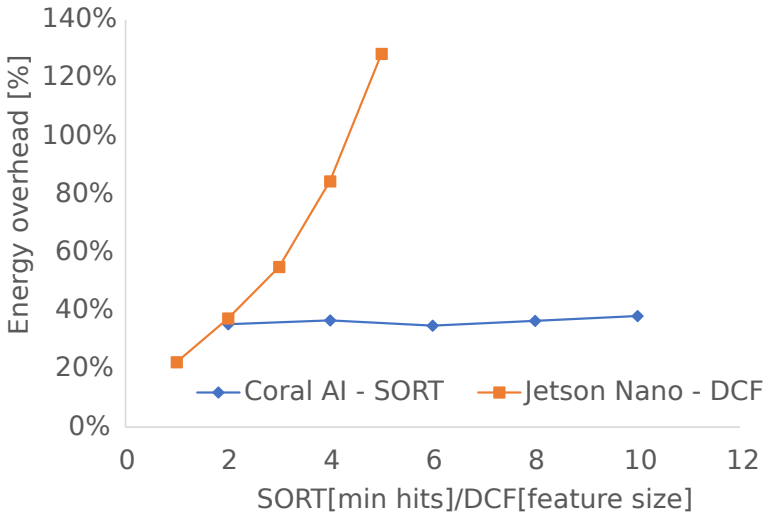


Figure 4.8: Energy overhead of the tracking algorithms when varying two representative parameters: *min hits* for SORT and *feature size* for NvDCF.

consumption of the SORT algorithm remains effectively constant regardless of the *min hits*

value. Since SORT relies primarily on Kalman filtering, which involves lightweight matrix operations on state vectors, changing the logic for track validation does not significantly alter the computational load per frame. Conversely, the NvDCF tracker exhibits a clear exponential dependency on the *feature size*. Unlike kinematic trackers, NvDCF processes raw pixel data to extract visual features for correlation. Increasing the size of the feature patch drastically increases the memory bandwidth usage and the number of floating-point operations required for each step.

4.4.2 Performance and Energy Saving

Having characterized the baseline tracking algorithms and their hardware energy costs, we now present the experimental results of the complete vision-based virtual sensor equipped with the Dynamic Inference Power Manager (DIPM). The primary objective of this analysis is not to evaluate the absolute accuracy of the underlying object detector, but rather to quantify the trade-off introduced by the DIPM. Specifically, how much energy can be saved versus how much tracking accuracy is sacrificed. We compare the virtual sensor's performance against the baseline results obtained using the same detection and tracking pipeline without any power management (i.e., processing every frame).

Detection and Tracking

To visualize the impact of the DIPM on the system's ability to maintain awareness of the scene, we logged the total number of unique objects actively tracked in each frame of the video benchmarks. Figure 4.9 reports the resulting log traces obtained with the DIPM disabled (solid blue line) and enabled (dashed red line) under three different parameter configurations. Specifically, the top plot corresponds to an aggressive power-management setting ($\alpha = 1$, $n_{max} = 6$), the middle plot to an intermediate configuration ($\alpha = 4$, $n_{max} = 6$), and the bottom plot to a strongly precautionary setup ($\alpha = 16$, $n_{max} = 6$). The vertical green bars indicate frames that were skipped by the DIPM. For skipped frames, no visual information is available; therefore, the trace is reconstructed by repeating the values from the last processed frame.

The overall trend shows that increasing the aggressiveness of the DIPM introduces a growing delay in object identification. By discarding frames, newly entering objects are detected and tracked later in time. Moreover, short-lived objects, visible as low-amplitude peaks in the blue baseline trace, may disappear entirely from the tracking results when the DIPM is active. In these cases, frame skipping shortens the effective track duration below the minimum threshold t_{min} required by the IOU tracker, leading to their rejection. Conversely, as the DIPM configuration becomes more conservative, the resulting trace progressively converges toward the reference behavior observed without frame skipping. From an energy-efficiency perspective, the traces highlight that a large number of frames are skipped during relatively static periods of the video, when few objects are present, or motion is slow. In contrast, during more dynamic phases characterized by faster movements or higher activity, the DIPM automatically reduces or suppresses frame skipping, thereby preserving tracking reliability when it is most needed.

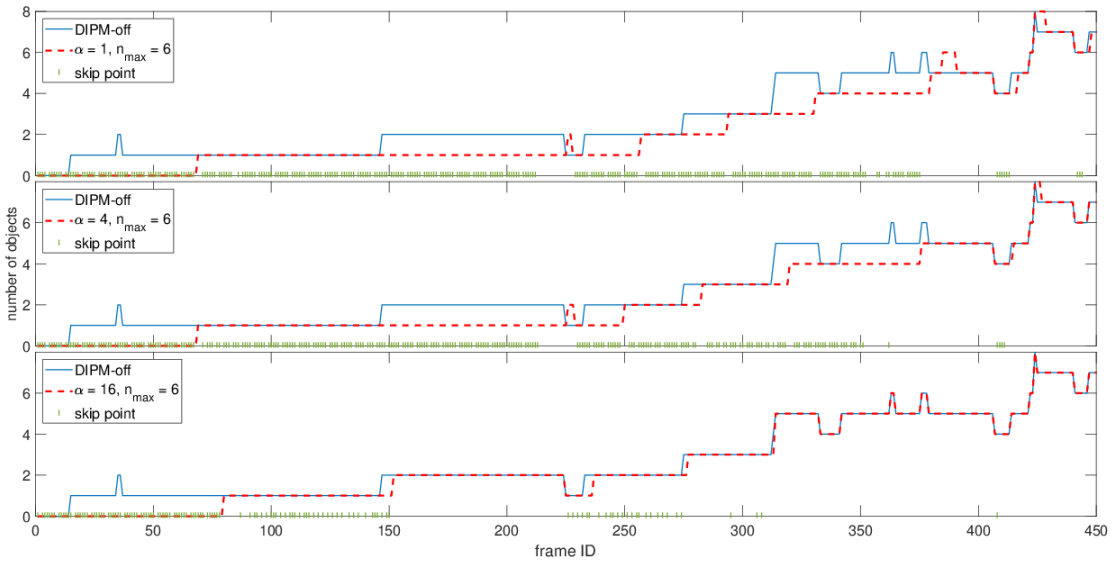


Figure 4.9: Log trace reporting the number of objects tracked per frame with and without DIPM. Green vertical bars identify frame skipping points.

MOT17 Results

The DIPM was evaluated on the MOT17 benchmark in order to rigorously quantify the trade-off between energy saving and tracking quality. We performed a sweep of the configuration space, varying the aggressiveness parameter α and the maximum skip limit n_{max}, and compared the results against the baseline tracker without power management (represented by a dashed red line) in Figure 4.10. The analysis reveals a clear relationship between the aggressiveness of the power manager and the fidelity of the tracking. Looking at the global accuracy metrics for HOTA and association precision (AssPr), we observe a rapid convergence towards the baseline performance as α increases. The parameter n_{max} acts as a safety net during this convergence. Lower values effectively bound the maximum error, preventing severe degradation even when the manager is aggressive. Most notably, the system identifies a stable operating point around $\alpha = 4$, where the performance gap for both metrics narrows to less than 1 percentage point regardless of the n_{max} setting. This implies that moderate frame skipping can be applied with negligible impact on the system's ability to correctly associate objects.

The mechanism behind the slight loss in accuracy is revealed by analyzing the error metrics in graphs (c) and (d). As the DIPM becomes more aggressive (lower α), we observe a corresponding rise in identity switches (IDSW) and track fragmentation (Frag). This behavior is physically consistent with the logic of the virtual sensor. Skipping frames increases the temporal gap between detections, which in turn increases the spatial displacement of moving objects. This larger displacement makes the association task more challenging for the IOU tracker, leading to occasional broken tracks or swapped identities. However, the global trend confirms the robustness of the proposed approach. Even in the

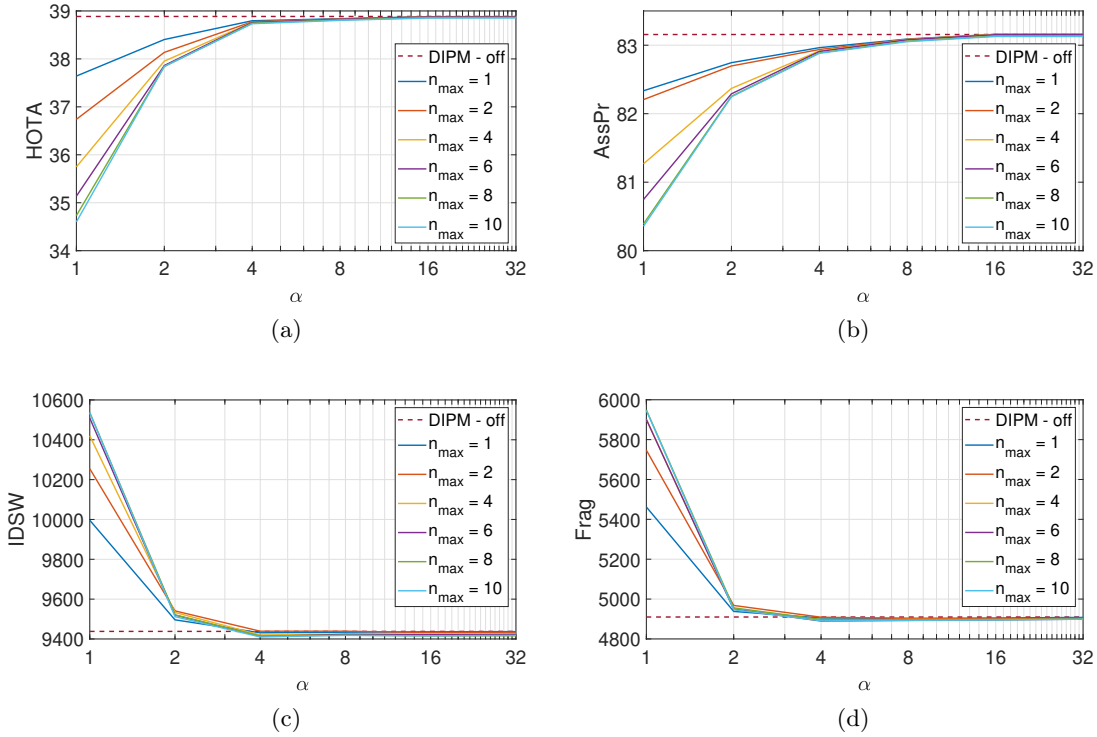


Figure 4.10: Results obtained on the MOT17 benchmark when changing the values of α and n_{max} related to the HOTA (a), association precision (b), identity switches (c), and tracks fragmentation (d).

worst-case scenarios (highly aggressive settings), the degradation of the primary metrics remains contained within 5%. Consequently, appropriately calibrated DIPM configurations allow the system to operate with substantial energy savings while maintaining a tracking quality statistically comparable to the full-frame baseline.

Impact of the DIPM on the Metrics Extraction

Beyond standard tracking metrics, it is crucial to verify how the DIPM affects the final output of the virtual sensor, namely, the occupancy statistics provided to the end-user. To this end, we measured the percentage error introduced by the power manager in estimating three key variables: the total count of unique objects, their average normalized distance, and their average normalized speed. The deviations were calculated on the two video benchmarks relative to the ground truth obtained with the DIPM disabled.

Figure 4.11 summarizes the results obtained on the benchmark containing vehicular traffic, reporting the error trends for the three metrics as a function of the DIPM parameters α and n_{max}. A consistent pattern emerges across all metrics. When α is close to 1, corresponding to a highly aggressive power-management strategy, the introduced error is significantly higher. As α increases, the DIPM becomes progressively more conservative,

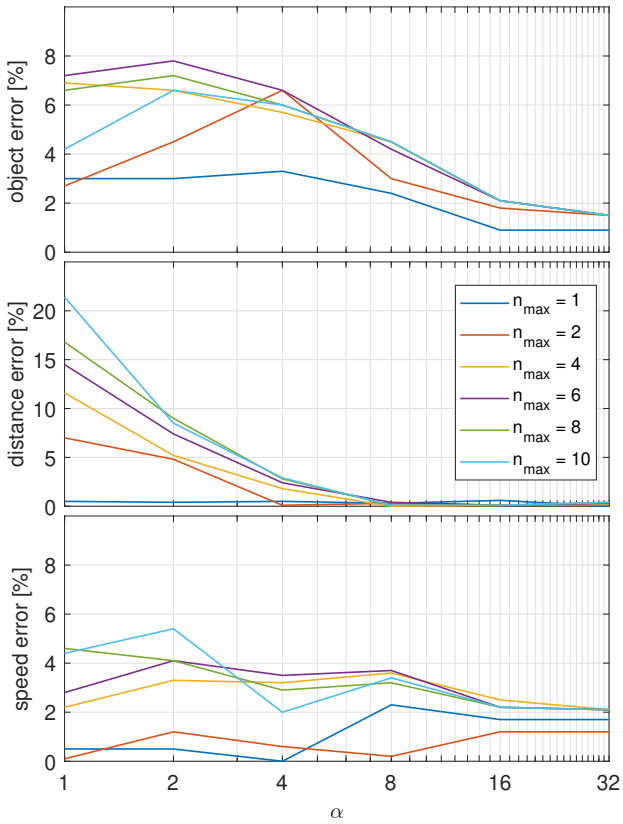


Figure 4.11: Plots showing the error introduced by the DIPM when changing the values of α and n_{max} in the calculation of the number of unique objects (top), normalized distance (middle), and normalized speed (bottom)

and the error rapidly decreases. This behavior reflects the role of α as a caution factor. Smaller values lead to a larger number of skipped frames, increasing the likelihood of missed associations and identity switches. These effects directly inflate the estimated number of unique objects and indirectly affect the computation of distance and speed by shortening or fragmenting object trajectories.

A similar effect is observed for the parameter n_{max}. Increasing its value allows the system to skip longer sequences of frames, extending the interval during which objects are not explicitly observed. This, however, raises the likelihood of losing tracks, which propagates into higher errors across all extracted metrics. The impact of n_{max} is therefore closely related to its role as a hard upper bound on how long objects can remain untracked without compromising metric reliability. Overall, the results show that although the DIPM introduces some distortion in the extracted metrics, this effect is highly controllable. By choosing moderate values for both α and n_{max}, the additional error can be limited to

a few percentage points, maintaining the reliability of the virtual sensor while achieving substantial reductions in computational load and energy consumption.

Energy Saving

Having validated the accuracy of the tracking algorithms and the precision of the metric extraction, the evaluation shifts to the most critical requirement for edge deployment: operational sustainability. This section presents the quantitative impact of the DIPM, demonstrating how the adaptive frame-skipping logic translates into tangible physical energy savings on the hardware. To validate the efficiency of the proposed solution, we measured the physical power consumption of the NVIDIA Jetson device using the measurement setup described in Section 3.3.5.

The tangible impact of the DIPM is immediately visible in the power profiles shown in Figure 4.12. The top trace represents the system under DIPM control, while the bottom trace shows the standard continuous tracking. In the standard operation (bottom

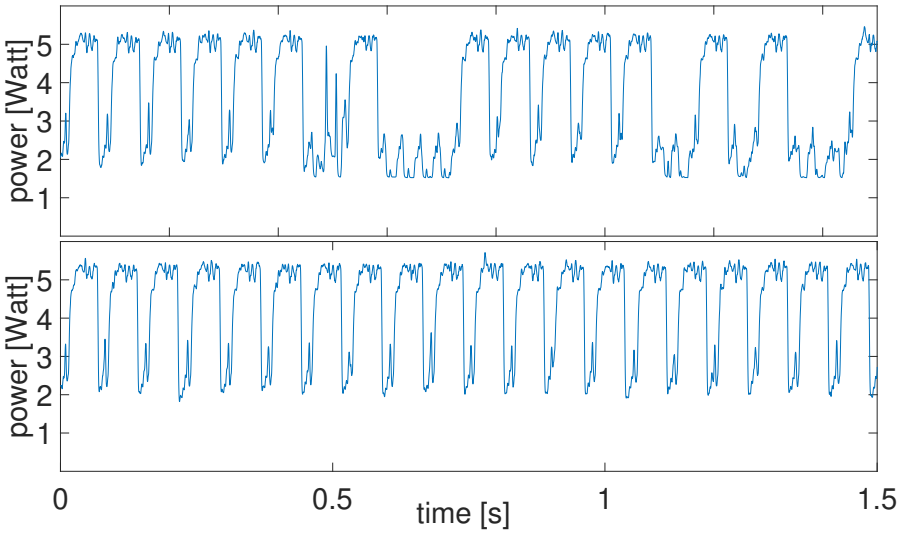


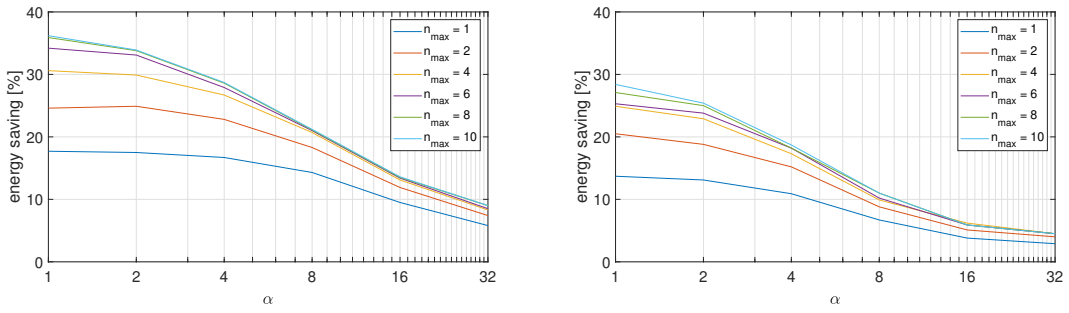
Figure 4.12: Power traces collected during object tracking with DIPM on (top trace) and off (bottom).

trace), each video frame triggers a computation burst peaking at approximately 5 Watts, corresponding to the GPU inference and tracking steps. In contrast, the trace with DIPM-enabled (top) reveals clear intervals of reduced activity. By skipping frames and bypassing the GPU inference entirely, the power consumption during these idle periods drops below 2 Watts, resulting in a net instantaneous gain of over 3 Watts per skipped frame. It is worth noting that this residual baseline consumption is due to the CPU continuing to decode the input video stream. While deeper savings could be achieved by disabling the webcam driver or video decoding at the OS level, we intentionally avoided such platform-dependent optimizations to demonstrate the general validity and portability of the proposed algorithmic

approach.

To evaluate how these instantaneous gains scale over long-term monitoring, we aggregated the total energy savings. However, the effectiveness of the DIPM is intrinsically linked to the dynamics of the environment. A quiet room allows for more frame skipping than a crowded hall. To capture this variability, the system was benchmarked against two distinct video scenarios representing common edge monitoring tasks: pedestrian surveillance and traffic monitoring. Both videos were captured using a fixed camera, providing the stable viewpoint typical of security installations. The first video source is a 120-second clip recorded at 30 fps, capturing pedestrian traffic on a sidewalk in the city center of Budapest, Hungary [309]. This sequence is characterized by slow-moving objects (people and cars) with non-linear trajectories and frequent occlusions. The second video is a 210-second clip recorded at 30 fps, showing vehicles traveling on Route 28 in West Dennis, Cape Cod, Massachusetts [310]. This sequence features faster-moving objects (cars and trucks) with linear trajectories and variable sizes.

Figure 4.13 illustrates the percentage of total energy saved for these two scenarios as a function of the DIPM parameters. Reducing the caution factor α or increasing the



(a) (b)

Figure 4.13: Percentage of energy savings obtained varying α and n_{max} for the video benchmark with cars (a) and people (b).

frame skip limit n_{max} results in greater energy savings. However, the extent of these savings is inherently dependent on the dynamics of the video content. For instance, the "cars" benchmark (a), which contains periods of low traffic density, enables the DIPM to capitalize on longer idle intervals. In contrast, the "people" benchmark (b), characterized by continuously moving targets and highly dynamic scenes, offers fewer opportunities for skipping frames, limiting potential energy reductions.

Finally, to identify the optimal configuration that maximizes energy savings while minimizing tracking error, we performed a Pareto analysis. Figure 4.14 plots the energy expenditure ratio (where lower indicates more savings) against the percentage error on normalized speed. The most efficient operating points, highlighted with red circles, are those closest to the origin, representing the best simultaneous minimization of cost and error. For the "cars" benchmark (a), the optimal trade-off is achieved with $n_{max} = 10$ and $\alpha = 2$. At this operating point, the virtual sensor achieves a remarkable 36% energy

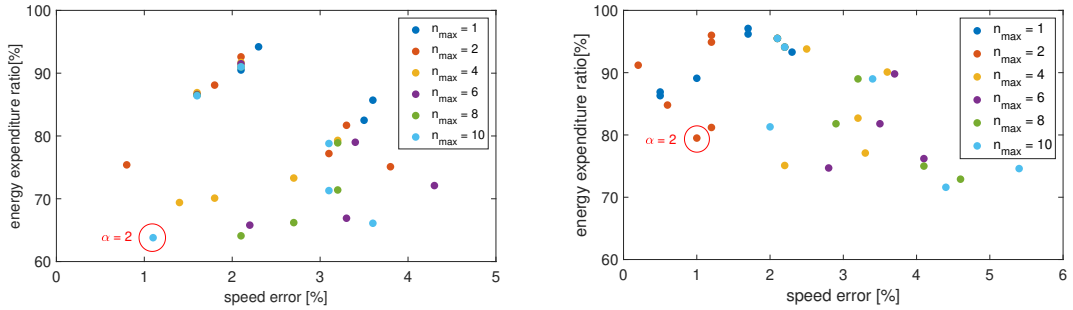


Figure 4.14: Pareto charts showing the energy expenditure rate versus the error on the normalized object speed for the video benchmark with cars (a) and people (b).

saving while maintaining an accuracy error of just above 1%. For the more challenging "people" benchmark (b), the optimal configuration is more conservative ($n_{max} = 2, \alpha = 2$), yet it still delivers a significant 21% energy saving with a similarly negligible error rate. These results demonstrate that the DIPM is effective not only in static scenarios with sporadic activity but also in dynamic environments. By adaptively modulating the inference rate, the system secures non-negligible energy efficiency gains without compromising the reliability of the extracted traffic metrics.

4.5 Summary

The approach discussed in this chapter reframes Computer Vision from a passive sensing modality into an active visual virtual sensor, capable of extracting high-level spatio-temporal descriptors, such as occupancy density, movement trajectories, and average speed, that can support adaptive sanitization strategies. In contrast to centralized surveillance solutions, this paradigm emphasizes privacy preservation and computational sustainability, as all processing is performed at the edge and only anonymized, aggregated information is transmitted.

The overall architecture is designed to operate within the strict resource constraints of embedded platforms like NVIDIA Jetson and Google Coral by relying on a lightweight tracking-by-detection pipeline. This design choice enables consistent estimation of motion-related metrics while maintaining robustness across varying scene conditions, without requiring complex geometric modeling or heavy post-processing.

The experimental validation confirmed that the choice of the tracking algorithm is of paramount importance. The analysis of the Intersection Over Union (IOU) tracker demonstrated its superiority for constrained environments. Compared to more computationally demanding alternatives such as NvDCF, the IOU-based solution increased energy consumption by only 9% relative to the detection stage alone. A multi-objective optimization on the MOT17 benchmark further identified a stable operating configuration that balances association precision and trajectory continuity, limiting identity switches while preserving

temporal coherence.

In addition, the integration of a Dynamic Inference Power Manager (DIPM) effectively decouples sensing frequency from constant power consumption by exploiting the temporal redundancy of static or low-dynamics scenes. By adapting the inference rate to the observed motion of targets, the system reduces instantaneous power usage from about 5 W to less than 2 W during idle phases. In realistic deployment scenarios, this adaptive strategy yields substantial energy savings, up to 36% in vehicular contexts and 21% in pedestrian environments, while keeping tracking performance metrics within a marginal degradation of approximately 5%.

Despite these promising results, the system's sensitivity to scene dynamics presents specific limitations. Primarily, the monocular 2D nature of the vision pipeline makes it inherently susceptible to perspective distortions and severe occlusions, which can occasionally affect the accuracy of spatial metrics in highly crowded scenes. Furthermore, the energy-saving potential of the DIPM is highly context-dependent. Its effectiveness peaks in environments characterized by frequent static or semi-static periods, such as offices and hospital corridors. Conversely, in constant high-traffic areas like transport hubs or large retail centers, the lack of idle times prevents the system from skipping inference frames, potentially reducing the energy savings to negligible levels.

Overall, these results indicate that the combination of lightweight vision algorithms with context-aware power management enables accurate and sustainable visual monitoring, making long-term deployment in smart indoor environments both practical and scalable.

Chapter 5

Sensor-Based Human Activity Recognition

The evolution of smart building management has traditionally focused on monitoring the environment through indirect metrics. Conventional systems often rely on IoT sensors and vision-based systems to infer cleaning needs based on usage load, such as triggering a maintenance request after a camera counts 100 people entering a room. However, human presence is not a uniform variable; the intensity of physical activity directly correlates with environmental degradation and biological risk. An individual running or moving vigorously through a space causes significantly more dust resuspension and bio-aerosol dispersion than someone remaining sedentary. Consequently, identifying the nature of occupant behavior is as critical as monitoring the air quality itself.

In the pursuit of biological safety and operational efficiency, Human Activity Recognition (HAR) represents the essential link between environmental dynamics and digital accountability. By shifting the focus from the room to the human agent, HAR provides a dual-layer of visibility: it allows the system to assess the real-time impact of occupants on their surroundings while simultaneously verifying the actual execution of hygiene protocols. By integrating wearable devices with Inertial Measurement Units (IMUs), this approach allows for objective validation of manual cleaning tasks, such as mopping, surface wiping, or hand-washing, while maintaining privacy in sensitive areas where video monitoring is restricted. This shift from mere presence detection to kinematic verification enables facility managers to align theoretical cleaning schedules with actual execution. Wearable devices thus become instruments that capture both the "how" and the "why" of sanitization, linking cleaning frequency directly to the behavioral patterns that drive environmental contamination.

Despite these advancements, a critical gap remains in the traceability of manual cleaning tasks and the high energy consumption required by wearable AI. Conventional monitoring can infer usage, but it cannot objectively verify if or how a sanitization protocol was executed. Furthermore, maintaining continuous HAR on wearable hardware often leads to rapid battery depletion, limiting the feasibility of long-term operational monitoring.

This chapter explores the technological layers required to transform raw inertial data

into actionable operational insights. Since continuous monitoring strains the limited battery resources of wearable hardware, the first challenge addressed is the optimization of the computational load. We propose a hierarchical trigger architecture designed to balance sensitivity and autonomy. This contribution balances analytical sensitivity with extreme energy autonomy by delegating initial detection to a low-power model. This trigger acts as a gatekeeper, activating the complex classifier only when relevant movement is detected, a strategy that significantly extends device lifespan by minimizing unnecessary processing. Once the activity is detected, the focus shifts to the system's ability to generalize across complex and variable cleaning tasks. Moving beyond standard classification, we introduce a metric learning framework based on Siamese Neural Networks. By distilling kinematic features into semantic templates, this approach allows the system to construct valid prototypes even for activities never encountered during training. This capability paves the way for Generalized Zero-Shot Activity Recognition (GZSAR), enabling the classification of unseen operational tasks without the need for extensive retraining. Finally, a further paradigm shift reinterprets physical activity not merely as signal data, but as a language. Just as words form sentences, sequences of kinematic patterns create complex operational narratives. This perspective naturally leads to the application of Transformer architectures and Large Language Models (LLMs), which are specifically designed to master the long-range dependencies typical of sequential data. We investigate whether these models can "speak" the language of inertial sensors to perform advanced tasks such as data imputation and synthetic data generation. However, the computational weight of these architectures poses a significant challenge, necessitating a rigorous evaluation of their feasibility on the energy-constrained embedded devices that define the boundaries of wearable HAR.

5.1 Background

Human Activity Recognition (HAR) is a research field focused on identifying the actions, goals, and intentions of one or more agents through the analysis of observations related to their movements and their surrounding environment [311]. Although HAR is now a pervasive technology, its origins date back to the early 1980s, when it emerged from the need to provide personalized assistance in domains such as healthcare, human–computer interaction (HCI), and sociology.

During the late 1980s and early 1990s, the research landscape in activity recognition was largely dominated by symbolic reasoning and external observation paradigms. In this context, Kautz introduced a logical theory of plan recognition, describing the process as logical inference based on a hierarchy of events. Attempting to deduce the "why" behind an action [312]. Parallelly, the Olivetti Research Laboratory pioneered the use of *Active Badges* to track and recognize the activities of multiple users in an office environment, representing one of the earliest attempts at wearable-assisted recognition [313]. As the limitations of purely symbolic approaches became evident, probabilistic reasoning and statistical learning methods began to gain traction. Notably, Charniak and Goldman proposed Bayesian frameworks capable of modeling the inherent uncertainty and variability of human behavior [314].

In this early phase, Computer Vision represented the dominant sensing medium for

HAR. Early approaches relied on handcrafted features extracted from video streams using techniques such as optical flow, motion history images, and background subtraction [315]. A major shift occurred in the early 2000s with the miniaturization and widespread adoption of Micro-Electro-Mechanical Systems (MEMS), which enabled the use of wearable inertial sensors. These devices made it possible to capture three-dimensional human motion without requiring fixed camera installations, thus broadening the applicability of HAR systems.

Modern HAR research can be broadly divided into two main strands according to the adopted data modality: (i) vision-based, and (ii) sensor-based. Vision-based HAR leverages RGB cameras to analyze images or video sequences and is particularly well-suited for monitoring multiple individuals or large crowds simultaneously. The introduction of depth cameras, such as the Microsoft Kinect, enabled accurate three-dimensional skeletal tracking, leading to significant performance improvements [316]. Despite these advantages, camera-based systems raise substantial privacy concerns, which severely limit their deployment in sensitive environments such as private homes, bathrooms, or changing rooms [317]. Moreover, their performance is often affected by environmental conditions, including poor illumination, occlusions, and constraints imposed by fixed camera viewpoints. Sensor-based HAR, in contrast, relies on time-series data collected from wearable devices equipped with accelerometers, gyroscopes, and magnetometers. This paradigm is inherently privacy-preserving, as it does not involve the acquisition of visual information about users or their surroundings. Additionally, wearable sensing enables continuous activity monitoring across diverse environments, independent of lighting conditions or indoor–outdoor settings. However, these approaches introduce their own limitations. Battery life remains a critical constraint, as wearable devices are typically characterized by limited energy resources [318].

The practical viability of sensor-based HAR has been significantly enhanced by the widespread adoption of wearable devices and the rapid advances in Artificial Intelligence. Recent years have marked the advent of what has been termed the "Era of Ubiquitous Motion Tracking" [319], in which smartwatches and fitness bands have evolved into powerful computing platforms. These devices are equipped with advanced IMUs and increasingly capable microcontroller units (MCUs) [64]. At the same time, the transition from manual feature engineering to Deep Learning has profoundly transformed the analysis of inertial sensor data. Whereas traditional machine learning techniques, such as Support Vector Machines (SVMs) and Decision Trees (DTs), depend heavily on handcrafted features and domain expertise, Deep Neural Networks (DNNs) are able to automatically learn rich and discriminative representations directly from raw sensor signals. Despite these advances, deploying state-of-the-art deep learning models directly on resource-constrained wearable devices remains a major open challenge, primarily due to stringent limitations in terms of energy consumption, memory, and computational capacity.

Problem Formulation

Sensor-based HAR is traditionally formulated as a supervised classification problem, where a predefined set of human activities is inferred from data collected by wearable or embedded

sensors, such as accelerometers, gyroscopes, and magnetometers. These sensors generate multivariate time-series signals that continuously describe the motion and orientation of the subject over time. A common pre-processing strategy in HAR consists of segmenting the continuous sensor streams into fixed-length temporal windows using a sliding window approach. Each window represents a short temporal snapshot of the performed activity and is treated as an independent sample to be classified. The window length and the degree of overlap between consecutive windows play a crucial role in the final performance of the system, as they directly affect both temporal resolution and feature stability. Prior studies have explored a wide range of window sizes, typically spanning from 1 to 30 seconds, highlighting the absence of a universally optimal configuration and the strong dependency on the nature of the activities and the sampling frequency of the sensors [320, 64, 321, 322, 323].

Formally, a single input sample can be represented as a multidimensional array of size $[n_s \times n_c]$, where n_s denotes the number of temporal samples within a window and n_c represents the number of sensor channels. Let $\mathcal{C} = c_1, c_2, \ldots, c_N$ be the set of N activity classes defined in the problem. The goal of a HAR system is to learn a function F that maps each input window to an activity label over time:

$$F([n_s \times n_c]_t) = \{c_1, c_2, ..., c_t\}, \qquad c_t \in \mathcal{C} \tag{5.1}$$

In conventional HAR pipelines, the function F is implemented using either shallow machine learning models, such as Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), or k-Nearest Neighbors (k-NN), or deep learning architectures, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). In all these cases, the task is framed as a closed-set multi-class classification problem, where the model is trained to assign each input sample to one of the known activity labels. For neural networks, this typically corresponds to a final layer with N output neurons, one per activity class, followed by a softmax operation that produces a probability distribution over the predefined classes. While effective, this formulation presents two major limitations. First, multi-class classifiers tend to increase in complexity as the number of activities grows, making them less suitable for deployment on resource-constrained devices such as wearables. Second, and more critically, traditional classifiers are inherently unable to recognize activities that were not present during training, a constraint that severely limits their adaptability in real-world scenarios.

To mitigate model complexity, binarization strategies have been widely adopted in HAR. These approaches decompose the original multi-class problem into a set of simpler binary classification tasks, which are generally easier to learn and more efficient to execute. Among the most common strategies are the one-vs-one (OVO) and one-vs-all (OVA) schemes [324]. In OVO, an N-class problem is split into $N(N-1)/2$ binary classifiers, each trained to discriminate between a specific pair of activities. During the training phase, each binary classifier is learned using only the samples belonging to the two target classes under consideration, while all instances associated with the remaining classes are excluded. This focused training setup often leads to simpler decision boundaries, as each classifier only needs to separate two activities with potentially well-defined differences. At inference

time, a test sample is evaluated by all pairwise classifiers. Each classifier produces a confidence score expressing the preference for one class over the other. These scores are then aggregated into a score matrix that captures the outcomes of all pairwise comparisons. The final activity label is determined through a confidence aggregation mechanism, typically by selecting the class that receives the highest number of wins or the maximum cumulative confidence across all pairwise decisions.

In the OVA, N binary classifiers are trained, each responsible for discriminating a single target activity from all remaining ones. During the training phase, all available samples are used. Instances belonging to the target class are labeled as positive, while samples from the other classes are treated as negative. At inference time, each classifier produces a confidence score indicating how likely the input window belongs to the corresponding activity. Collectively, these outputs form a score vector at time t:

$$R_t = (r_1, r_2, ..., r_i, ...r_N) \tag{5.2}$$

where $r_i \in [0, 1]$ represents the confidence score associated with class c_i. The final activity prediction is obtained by selecting the class with the highest confidence:

$$c_t = argmax(R_t) \tag{5.3}$$

Although the OVA strategy simplifies the learning task compared to a full multi-class classifier, it still assumes that all possible activities are known in advance and explicitly represented during training. As a consequence, the system remains inherently closed-set and is unable to generalize to unseen activities. This limitation motivates a reformulation of the HAR problem from direct classification to similarity learning.

Zero-Shot Activity Recognition

Zero-Shot Activity Recognition (ZSAR) addresses the problem of recognizing human activities that are not available during the training phase. In this setting, the set of activity classes is divided into two disjoint subsets: the *seen classes*, denoted as $\mathcal{S}$, which are used to train the model, and the *unseen classes*, denoted as $\mathcal{U}$, which appear only at test time. Formally, the complete label space is defined as $\mathcal{C} = \mathcal{S} \cup \mathcal{U}$, with $\mathcal{S} \cap \mathcal{U} = \varnothing$.

Because no labeled data are available for unseen activity classes during training, conventional supervised classification cannot be applied directly. Zero-Shot Activity Recognition (ZSAR) overcomes this by leveraging auxiliary semantic information that creates a shared representation between seen and unseen activities. This semantic information can take the form of manually defined attributes, textual descriptions, or embedding vectors encoding high-level relationships among activities. The learning task is thus reframed: rather than mapping sensor inputs directly to class labels, the system learns to associate sensor patterns with the corresponding semantic representations. In its original formulation, ZSAR assumes that all test samples belong solely to unseen classes. While this simplifies evaluation, it rarely reflects real-world conditions. In practical HAR deployments—particularly with wearable or ambient sensors—streams contain a mixture of known and unknown activities. Limiting the test set to unseen classes alone reduces the operational relevance of

standard ZSAR methods.

To address this limitation, the concept of Generalized Zero-Shot Activity Recognition (GZSAR) has been introduced. In GZSAR, test instances may belong to either seen or unseen classes, making the recognition task significantly more challenging but also more representative of real deployments. The model must therefore be capable of correctly classifying known activities while simultaneously identifying activities it has never been explicitly trained on. A major challenge in GZSAR is the so-called seen-class bias. Because the model is trained only on data from seen classes, it tends to assign higher confidence scores to them during inference. As a result, samples from unseen classes are often incorrectly classified as belonging to one of the seen classes. This bias stems from the stronger statistical attraction learned for seen classes compared to unseen ones, whose representations are only indirectly available through semantic information. To mitigate this issue, calibration strategies have been proposed in the literature. One notable approach, introduced by Changpinyo et al., employs a calibration factor to explicitly rebalance the confidence assigned to seen and unseen classes during inference [325]. By scaling the scores associated with seen classes using a calibration parameter α, the classifier can reduce its dominance and improve recognition performance on unseen activities. This mechanism plays a crucial role in making GZSAR viable for practical HAR systems, where both reliability on known activities and sensitivity to novel behaviors are required.

5.1.1 Wearable Technologies for HAR

The widespread adoption of sensor-based HAR has been catalysed by the exponential growth of the wearable technology market. Devices that were once considered niche gadgets have evolved into essential everyday tools, ranging from lightweight fitness bands to advanced smartwatches. This transition has been driven by improvements in battery life, reliable wireless connectivity, and, most crucially, the integration of low-power MEMS designed for continuous health and fitness monitoring.

Market analyses project that the global smartwatch sector will exceed $96 billion by 2027 [326], effectively establishing a large-scale, distributed sensing infrastructure. At the core of this ecosystem lies the IMU. Modern wearable devices typically integrate a tri-axial accelerometer that is capable of measuring proper acceleration and is ideal for detecting static postures or periodic locomotion. This sensor is often paired with a tri-axial gyroscope, which captures angular velocity and is essential for recognizing fine-grained rotational movements. High-end devices frequently complement these with magnetometers for orientation correction, as well as biosensors like Photoplethysmography (PPG) or Electrodermal Activity (EDA) sensors [327, 328].

The availability of such hardware has profoundly influenced the architectural design of HAR systems. Early solutions treated wearables merely as passive data collectors, streaming raw measurements to a smartphone or cloud backend. However, the contemporary approach, often referred to as Edge AI or TinyML, advocates for moving the intelligence directly to the data source. In this context, three main architectural paradigms can be identified [118, 121]. The cloud-centric architecture relies on transmitting raw data to remote servers. While this offers virtually unlimited computational resources, it suffers from

high latency, privacy risks, and the significant energy cost of continuous wireless transmission. The smartphone-tethered architecture offloads computation to a paired mobile device. This reduces latency but necessitates a persistent connection, draining the batteries of both devices. The on-device architecture performs sensing, feature extraction, and inference entirely on the wearable's MCU. This last approach offers distinct advantages that make it particularly suitable for real-world deployment. First, privacy-by-design is ensured, as raw sensor data, which could reveal sensitive behavioral patterns, never leaves the device. Second, latency is minimized, enabling immediate feedback to the user (i.e., a haptic alert upon task completion). Lastly, the system guarantees autonomy, operating independently of internet access or unstable Bluetooth connections.

This architecture is particularly relevant for the specific domain of cleaning and sanitization monitoring. Unlike environmental sensors or cameras, which can be obstructed or raise privacy concerns in sensitive areas like restrooms or hospital wards, a wrist-worn device is unobtrusive and privacy-preserving. Furthermore, since cleaning activities such as mopping, wiping, or vacuuming are predominantly manual, a sensor located on the wrist can capture the unique kinematic signature of these actions with significantly higher fidelity than a smartphone in a pocket. Recent studies have demonstrated the viability of this approach by running Deep Learning models directly on smartwatches to recognize hand-washing gestures in real-time, enabling objective verification of hygiene compliance without invasive surveillance [40, 329].

On-device HAR faces several constraints. Wearable MCUs typically have limited memory and processing capacity, restricting the size and complexity of deployable models. Additionally, although modern sensors and processors are increasingly efficient, the overall energy budget remains a critical bottleneck for continuous, long-term monitoring.

5.1.2 Energy Constraints in Continuous Wearable Monitoring

While the capabilities of wearable devices have expanded, their practical utility in HAR is fundamentally limited by energy availability. Modern smartwatches typically rely on small Lithium-Ion batteries with capacities ranging from 300 to 450 mAh. Unlike smartphones, which are recharged daily, users expect wearables to last multiple days. However, the continuous sampling of inertial sensors combined with the processing of this data can deplete the battery in a matter of hours if not managed meticulously.

A major source of energy inefficiency in HAR applications stems from the discrepancy between the monitoring time and the event duration. Applications such as sanitization monitoring focus on sporadic events (i.e., a cleaner washing their hands or wiping a surface) that may occur only a few times per hour. Yet, to detect them, the device must remain active and process sensor data 100% of the time. In the domain of low-power telecommunications, this phenomenon is known as "overhearing", where a device wastes energy processing data not intended for it [330]. To mitigate this, HAR architectures increasingly adopt hierarchical triggers. This concept is analogous to keyword spotting in voice assistants, where a low-power hardware listens for activation commands like "Hey Siri" or "Alexa". In this setup, a lightweight, ultra-low-power algorithm runs continuously

to detect a broad motion signature. Only when this signature is detected is the energy-intensive Deep Learning model activated to perform fine-grained classification [141, 140].

Beyond architectural patterns, energy efficiency must be addressed at the software level. Standard smartwatch operating systems like Wear OS are often not optimized for continuous high-frequency sampling, leading to significant overhead from background services and unoptimized I/O operations [331]. To address this, specialized libraries have been developed [332, 333]. These libraries enable energy-aware data collection by buffering sensor data while keeping the device's main processor in a low-power *doze* mode for as long as possible, significantly extending battery life during active sessions.

Shifting HAR to Edge AI requires balancing computation and communication costs. Offloading raw sensor data to the cloud reduces local computation but incurs high radio energy consumption over Bluetooth or Wi-Fi. Local inference, by contrast, consumes CPU energy but transmits only the final classification, often proving more energy-efficient for continuous monitoring when models are optimized [122, 123].

Adaptive inference techniques further improve this trade-off. Early-exit neural networks [143] allow computation to terminate for simple samples, while dynamic offloading strategies [140, 142] switch between local and cloud processing based on confidence. These optimizations are critical to enabling wearable HAR systems that are both accurate and energy-efficient.

5.1.3 Transformer and Large Language Models in HAR

Traditional sensor-based HAR relies on classical machine learning methods like SVMs, Decision Trees, k-Nearest Neighbors, or deep learning models such as CNNs, RNNs, and LSTMs. These approaches perform adequately but often struggle to capture long-range temporal dependencies and subtle correlations in sequential sensor data, particularly for activities spanning extended periods. Recently, sequence modeling has been transformed by Transformers and Large Language Models (LLMs), which leverage attention mechanisms to model temporal dependencies more effectively. Unlike recurrent architectures, Transformers can capture global context across entire sequences, enabling the recognition of complex and temporally extended human activities with higher fidelity.

Originally introduced by Vaswani et al. [334] for Natural Language Processing (NLP), Transformer models rely entirely on a mechanism called *self-attention*. This allows the model to weigh the significance of distinct time steps simultaneously, enabling the network to correlate disparate signal segments like the beginning and end of a complex gesture, regardless of their temporal distance. The three core components of the Transformer architecture are: (i) self-attention, (ii) positional encoding, and (iii) residual connections. The self-attention mechanism enables the model to capture relationships between all elements of an input sequence by computing a weighted combination of feature representations. Given a query vector Q, a set of key vectors K, and value vectors V, self-attention produces an output as a weighted sum of the values, where the weights are determined by a compatibility function between queries and keys. Both multiplicative (dot-product) and additive attention mechanisms are commonly used. An extension of this mechanism, known as multi-head attention, allows the model to attend to multiple representation subspaces

simultaneously by performing several attention operations in parallel [334]. Since the self-attention mechanism is inherently invariant to sequence order, positional encodings must be injected into the model input to preserve the temporal structure of the data [335]. In the original Transformer formulation, fixed sinusoidal encodings are used to provide unique representations for each position while preserving relative temporal relationships. Residual connections, also referred to as skip connections, were originally introduced in deep CNN architectures such as ResNet [336]. They allow the input of a layer to be added directly to its output, mitigating the vanishing gradient problem and facilitating the training of deep architectures. Formally, they are expressed as:

$$output = layer(x) + x \tag{5.4}$$

where x denotes the layer input. Together, these architectural components allow Transformers to effectively model complex temporal dependencies in sensor data, outperforming LSTMs in capturing the global context of motion sequences [337, 338].

Building on this architecture, LLMs like Bidirectional Encoder Representations from Transformers (BERT) have demonstrated that vast amounts of knowledge can be encoded into models through self-supervised pre-training [339]. An emerging research question is whether these capabilities can be transferred to HAR. Researchers have successfully applied LLM techniques to time-series tasks by tokenizing accelerometer and gyroscope data and treating distinct motion patterns as "words". In this context, data imputation is a relevant application. Using techniques like Masked Language Modeling (MLM) to help models like BERT reconstruct missing segments of sensor data with higher fidelity than statistical interpolation [340, 341]. Another application of LLMs is synthetic data generation. Models can be trained to generate realistic sensor traces, addressing the chronic scarcity of labeled HAR datasets [342, 343].

While the theoretical advantages of Transformers are clear, their practical application in wearable HAR faces a significant hurdle due to computational cost. Transformers are notoriously resource-intensive, requiring substantial memory and processing power that far exceed the capabilities of standard MCUs. Research into Tiny Transformers is still in its infancy. Some works have proposed lightweight variants or hybrid CNN-Transformer architectures deployed on edge platforms like NVIDIA Jetson [344, 345]. However, deployment on ultra-low-power devices remains an open challenge. Therefore, bridging the gap between the reasoning capabilities of LLMs and the constrained reality of wearable hardware represents a critical frontier in current HAR research.

5.1.4 Siamese Neural Networks and Deep Metric Learning

In traditional Deep Learning, classification tasks are typically addressed by training a network to map an input sample to one of a set of predefined classes, for instance, through a final Softmax layer. However, this paradigm exhibits significant limitations in dynamic scenarios, where the number of classes may change over time, or when labeled data are scarce, as in few-shot learning or in the recognition of previously unseen activities (zero-shot learning). To overcome these limitations, the literature has introduced Siamese Neural

Networks (SNNs). An SNN is not merely a specific network architecture, but rather a learning paradigm centered on similarity instead of direct classification. Its primary objective is not to assign a class label to an input, but to learn a mapping function that projects data into a latent space (embedding space) in which the geometric distance between two points reflects their semantic similarity [96].

The term *siamese* originates from the intrinsic structure of the model, which consists of two or more identical subnetworks. These subnetworks share the same architecture, the same configuration, and, most importantly, the same parameters through a mechanism known as weight sharing. This property ensures that two identical inputs, processed by the two branches, are mapped to the same point in the feature space. As illustrated in Figure 5.1, a basic SNN architecture is composed of three functional components: (i) feature extractor, (ii) comparison head, and (iii) decision-making head.

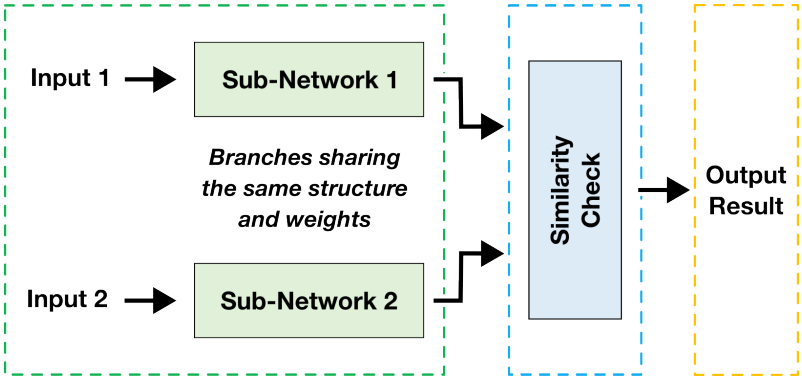


Figure 5.1: Structure of a Basic Siamese Neural Network: the feature extractor (dashed green line), the comparison body (dashed blue line), and the decision-making head (dashed yellow line).

The feature extractor is the core of each branch. Its purpose is to process the input and produce a low-dimensional feature vector. Depending on the nature of the signal, this block may be implemented using Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, or other deep architectures. The comparison head is the stage where the two branches converge. A distance metric is computed between the two feature vectors produced by the encoders. Commonly used metrics include the Euclidean distance and cosine similarity. Finally, the decision-making head is the final layer that transforms the computed distance into a similarity score. The score is often expressed as a probabilistic value between 0 and 1, indicating whether the two inputs are likely to belong to the same class.

Unlike conventional neural networks, which learn from individual samples, SNNs are trained using pairs of samples. The training dataset must therefore be reformulated into pairs, each associated with a binary label:

$$Y = \begin{cases} 1 & \text{if } X_1 \text{ and } X_2 \text{ belong to the same class (positive pair)} \\ 0 & \text{if } X_1 \text{ and } X_2 \text{ belong to different classes (negative pair)} \end{cases} \tag{5.5}$$

Model optimization relies on specialized loss functions such as Contrastive Loss or Triplet Loss. The objective of these losses is to minimize the distance between positive pairs, thereby pulling similar embeddings closer together, while maximizing the distance between negative pairs, pushing dissimilar embeddings apart. A critical aspect of this training phase is pair mining. Since the number of possible negative pairs grows combinatorially and largely exceeds the number of positive pairs, balanced sampling strategies are required to prevent the network from converging to trivial solutions, such as always predicting dissimilarity.

Although SNNs were originally introduced for signature verification and later became widely adopted in computer vision tasks such as face recognition, they have also proven highly effective in sensor-based HAR [346]. Traditional HAR approaches often suffer from limited generalization across different users, a phenomenon commonly referred to as subject dependency. A model trained on the movement patterns of a given subject frequently performs poorly when applied to another individual, due to intrinsic biomechanical variability. Similarity-based HAR addresses this issue by learning a more user-invariant distance metric. Rather than modeling the exact execution of a specific activity, an SNN learns how similar two activity instances are, independently of the person performing them. Several studies have shown that this paradigm enables efficient personalization with very limited data, supporting few-shot learning scenarios [163, 347]. In practice, a single example of an activity performed by a new user can be sufficient for the system to recognize future occurrences by comparing incoming signals against that reference example.

The use of SNN goes beyond the direct comparison of raw sample pairs, forming the foundation for a more advanced concept named semantic template generation. In most HAR literature, classification is performed by comparing an unknown input against a large set of labeled samples, for example, using k-Nearest Neighbors. While effective, this approach is computationally expensive and poorly suited to resource-constrained devices. The key insight is that SNNs can be exploited to distill the information of an entire class into a compact prototype or template. Because the network projects data into a coherent semantic space, it becomes possible to aggregate the feature representations of multiple executions of the same activity (i.e., hand washing) into a single representative vector, referred to as a semantic template. During inference, the system no longer performs traditional multi-class classification. Instead, it measures the distance between the incoming sample and the available templates. This approach naturally enables GZSAR scenarios. If a valid semantic template is available, the system can recognize activities that were not present in the original training set, relying solely on geometric similarity in the learned feature space.

5.2 Research Contribution

The transition from theoretical activity recognition to operational sanitization systems demands a rigorous methodology that balances classification fidelity with the strict energy constraints of wearable hardware. To move beyond static cleaning schedules toward dynamic, process-verified management, intelligent wearable nodes must be capable of processing complex kinematic data directly at the Edge. The following sections detail the

technical architecture designed to bridge the gap between raw inertial streams and actionable operational insights. This framework first addresses the energy bottleneck through a hierarchical trigger architecture, which intelligently gates the activation of complex classifiers to minimize redundant processing. To ensure adaptability in dynamic environments, the system leverages Siamese Neural Networks to distill activities into semantic templates, enabling the recognition of unseen cleaning tasks without extensive retraining. Finally, the methodology reinterprets human motion as a sequential language, applying Transformer architectures and LLMs to explore advanced generative capabilities while rigorously benchmarking their feasibility on resource-constrained embedded devices.

5.2.1 The Lightweight Accurate Trigger

Deploying a comprehensive HAR system on wearable devices immediately confronts the harsh reality of energy constraints. While deep learning models offer the precision required to distinguish complex sanitization tasks, running them continuously is computationally prohibitive for battery-powered hardware. This is particularly critical in professional cleaning scenarios, where relevant operational activities are interspersed with long periods of low-relevance movement like walking, standing, or idling. To address this inefficiency, we move away from the unsustainable always-on paradigm in favor of a hierarchical control logic. The foundation of this approach is the Lightweight Accurate Trigger (LAT). By formalizing the problem of activity detection not as a simple energy threshold but as a distinct binary classification task, the LAT optimizes the delicate balance between recognition fidelity and power consumption. This ensures that the heavy computational resources of the primary classifier are mobilized only when a relevant operational gesture is actually occurring, thereby extending the device's operational lifespan without sacrificing the capture of critical events.

Problem Formulation

Sensor-based HAR aims to map a continuous stream of raw sensory data, such as accelerometer, gyroscope, and magnetometer, to a discrete set of semantic labels. In a real-time continuous monitoring context, the system processes an input stream s_t to detect and characterize specific behaviors chosen from the range of activities a person might perform daily. For instance, a step counter continuously monitors accelerometer signals, searching specifically for human step signatures to count them. Let $A = \{a_1, a_2, \ldots, a_N\}$ denote the set of N predefined target activities. The standard HAR objective is to construct a classification function F that predicts the sequence of activities based on the input signal:

$$F(s_t) = \{a_1, a_2, ..., a_t\}, a_t \in A \tag{5.6}$$

However, deployment in real-world environments introduces the open set recognition problem. The set of activities performed by a user in daily life, denoted by $\mathcal{M}$, inevitably supersedes the limited subset of trained activities A (i.e., $A \subset \mathcal{M}$) [348]. Without an additional handling mechanism, these unknown activities are inevitably matched to one of the available classes in A, leading to misclassifications. To address this, a more robust

approach introduces the concept of Unknown Activities (UAs). In this framework, if the prediction confidence level for any known class falls below a certain threshold, the system returns the *unknown* class, indicating that none of the target activities are present in the input data. Adding the this class rewrites Equation 5.6 as:

$$F(s_t) = \begin{cases} \{a_1, a_2, ..., a_t\}, a_t \in A, if\ max(c(a_t)) > th \\ \{unknown\}, otherwise \end{cases} \tag{5.7}$$

Where $\max(c(a_t))$ represents the maximum classification confidence value for each activity $\in A$.

While this approach effectively mitigates incorrect classifications, it still incurs a large consumption of computing and energy resources even when the input does not contain an activity of interest (i.e., when $F(s_t) \notin A$). This is because the full classifier must process every sample to determine the confidence level. Starting from these premises, we implement a trigger system that initiates the classification procedure only when one of the known activities is present in the input signal. Ideally, this trigger system acts as a binary classifier capable of distinguishing, with maximum accuracy, whether the input data belongs to the subset of interesting activities (A) or the class of unknown activities (UAs). Crucially, this system must be computationally lightweight to minimize energy consumption; we name it Lightweight Accurate Trigger (LAT).

Energy Saving

To quantify the potential efficiency gains of the architecture, we first consider an ideal LAT implementation that uses a binary classifier with 100% accuracy. In this theoretical upper bound, the system's efficiency is strictly a function of the consumption ratio between the trigger and the baseline (full classifier), and the probability that any of the target activities is present in the input signal $p(A)$. The effective classification energy (ECE), derived from the energy consumed by the LAT (E_{LAT}) and the baseline classifier (E_{BASE}), is defined as:

$$ECE = p(A) \times (E_{LAT} + E_{BASE}) + (1 - p(A)) \times E_{LAT} \tag{5.8}$$

Figure 5.2 visualizes the theoretical energy savings relative to the baseline as a function of the ratio E_{BASE}/E_{LAT} across varying probabilities $p(A)$. As anticipated, substantial energy conservation is achievable when the prevalence of target activities is minimal. For significant gains, the baseline energy consumption must exceed that of the trigger by at least one order of magnitude. Specifically, with a LAT consuming only one-tenth of the baseline energy ($E_{LAT} = \frac{1}{10}E_{BASE}$), savings reach up to 55% when target activity probability is $p(A) \leq 0.3$. Transitioning to a real-world scenario necessitates the incorporation of classification stochasticity. The LAT output regarding the target set A falls into four categories: true positive (TP), true negative (TN), false positive (FP), or false negative (FN). Each outcome has a distinct energetic implication. If a TP occurs, the system correctly identifies a target activity and triggers the complete classifier. In this case, the energy cost is cumulative ($E_{BASE} + E_{LAT}$). On the other hand, in the case of TN, the system correctly identifies a non-target activity. The complete classifier remains dormant,

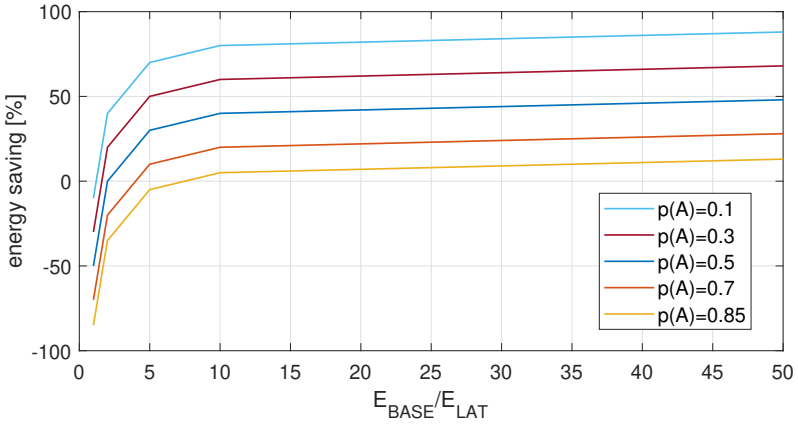


Figure 5.2: Theoretical energy saved by the triggered approach with respect to the baseline (i.e., by using only the complete classifier) when varying the E_{BASE}/E_{LAT} ratio for different probability $p(A)$

yielding the maximum energy saving (E_{LAT} only). When a FP occurs, the system erroneously triggers the complete classifier. This represents a missed opportunity for saving, incurring the full cumulative cost ($E_{BASE} + E_{LAT}$) unnecessarily. Finally, in the case of FN, the system fails to trigger for a target activity. While this reduces energy consumption to E_{LAT}, it represents a critical failure in recognition performance. Incorporating these non-idealities, Equation 5.8 becomes:

$$\begin{aligned} ECE = &\Big[p(A) \times TPR + (1 - p(A)) \times FPR\Big] \times (E_{LAT} + E_{BASE}) + \\ &\Big[(1 - p(A)) \times TNR + p(A) \times FNR\Big] \times E_{LAT} \end{aligned} \tag{5.9}$$

Where TPR, FPR, TNR, and FNR represent the rates of the corresponding output states. Leveraging the complementary nature of these rates ($TPR = 1 - FNR$ and $TNR = 1 - FPR$), we reformulate Equation 5.9 to isolate only FPR and FNR:

$$\begin{aligned} ECE = &\Big[p(A) \times (1 - FNR) + (1 - p(A)) \times FPR\Big] \times (E_{LAT} + E_{BASE}) + \\ &\Big[(1 - p(A)) \times (1 - FPR) + p(A) \times FNR\Big] \times E_{LAT} \end{aligned} \tag{5.10}$$

Figure 5.3 illustrates the impact of the FPR on energy savings. Because every FP invokes the computationally intensive baseline, the energy penalty scales directly with the FPR. This waste is exacerbated at lower values of $p(A)$ as the sparse occurrence of target activities implies a high volume of negative samples, thereby increasing the absolute number of false triggers. Conversely, Figure 5.4 depicts the relationship between energy savings and the FNR. While the data indicates that increasing FNR reduces energy consumption by preventing the baseline from waking, this is a false economy. Each FN corresponds to an unrecoverable failure of the classification system to detect a user's activity. Therefore,

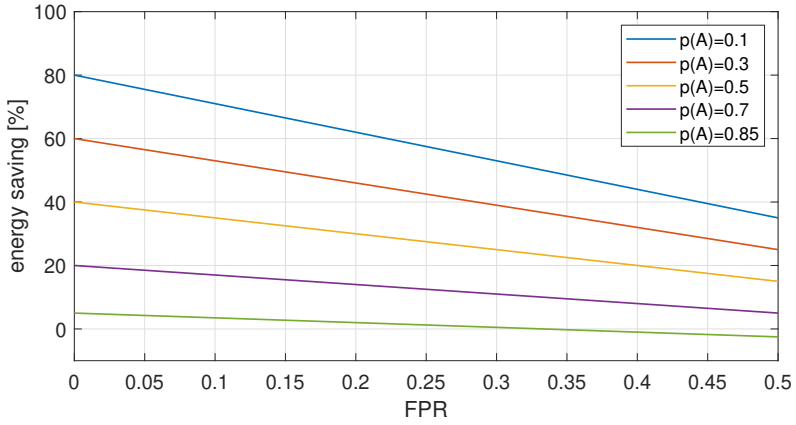


Figure 5.3: Theoretical energy saved by the triggered approach with respect to the baseline when varying the FPR for different values of the probability $p(A)$

energy reduction via FNR is not a viable optimization strategy.

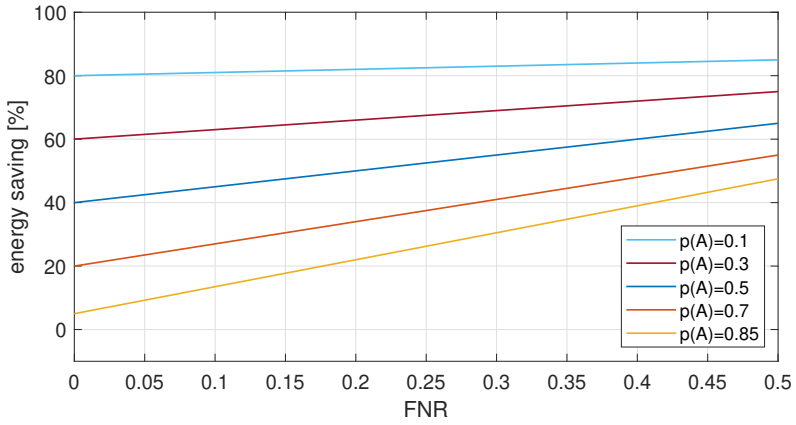


Figure 5.4: Theoretical energy saved by the triggered approach with respect to the baseline when varying the FNR for different values of the probability $p(A)$

Consequently, for this LAT architecture, in order to deliver significant energy savings without compromising system utility, three conditions must be simultaneously satisfied:

$$E_{BASE} \gg E_{LAT} \tag{5.11}$$

$$p(A) \ll 1 \tag{5.12}$$

$$FPR = as\ low\ as\ possible \tag{5.13}$$

Classification Performances

A fundamental prerequisite for the deployment of the LAT framework is that the architectural split must not degrade the recognition fidelity of the system. Ideally, the overall misclassification rate of the triggered system (MCR_{system}) should remain comparable to that of the standalone baseline (MCR_{base}). In this architecture, classification errors can propagate through three distinct pathways:

- The LAT erroneously classifies a true input as false (FN). Consequently, the complete classifier is not triggered, and the activity is missed entirely.
- The LAT correctly identifies a positive input (TP) and triggers the baseline, but the baseline subsequently misclassifies the specific activity label.
- The LAT erroneously classifies a false input (FP) and triggers the baseline, but it misclassifies the input.

Mathematically, the total misclassification rate of the proposed system can be formalized as:

$$MCR_{system} = FNR + TPR \times MCR_{base} + FPR \times MCR_{base} \tag{5.14}$$

Where FNR, TPR, and FPR denote the performance metrics of the LAT, and MCR_{base} represents the intrinsic error rate of the baseline model. By grouping the terms associated with the baseline's activation, we can rewrite Equation 5.14 as:

$$MCR_{system} = FNR + (TPR + FPR) \times MCR_{base} \tag{5.15}$$

This formulation reveals a critical insight. The system's error profile is dominated by the FNR of the trigger. The term $(TPR + FPR) \times MCR_{base}$ represents errors that occur only when the baseline is active. In the case of a TP, the error rate is bounded by the baseline's high performance. In the case of an FP, the robust baseline acts as a secondary filter, likely rejecting the unknown input, thereby mitigating the LAT's error. Conversely, an FN represents an irrecoverable loss of information; the baseline is never consulted, and the system fails immediately. Therefore, the fourth and final design requirement for the LAT is:

$$FNR = as\ low\ as\ possible \tag{5.16}$$

Synthesizing the four requirements derived above delineates the operational boundaries and design challenges of the approach. The requirements on classification performance, in particular the need to keep both FPR and FNR low (Eq. 5.13 and Eq. 5.16), impose clear constraints on the LAT architecture. The model must have enough expressive capacity to reliably distinguish the target activities from background noise. At the same time, this level of accuracy conflicts with the energy constraint (Eq. 5.11), which requires the computational cost of the LAT to remain minimal with respect to the baseline system. As a result, the design of the LAT can be seen as an optimization problem, where the architecture must be sufficiently accurate to be dependable, but also lightweight enough to ensure overall energy savings. In addition, the probability constraint (Eq. 5.12) defines the intended application domain of this approach. The proposed topology is suitable for

scenarios in which the target activities occur infrequently. If the probability of occurrence is high, the baseline system would be activated too often, and the advantages provided by the trigger mechanism would be largely lost, making the architecture inefficient.

Machine Learning Models

The design of a LAT for sensor-based HAR can follow two main approaches: using traditional signal processing with template matching or adopting a machine learning-based methodology. Template matching relies on identifying characteristic signal *signatures* for each activity based on a detailed physical understanding. While effective for simple movements, this approach is rigid and struggles to accommodate multiple complex activities within a single trigger. The machine learning approach provides greater flexibility. Here, a binary classifier is trained directly on accelerometer and gyroscope streams to distinguish target activities from background or idle motion. This method automatically learns relevant patterns, enabling robust detection across a variety of movements.

Regarding neural architectures, CNNs, LSTMs, and hybrid CNN-LSTM models are particularly suitable. CNNs effectively extract spatial features from the raw sensor signals, while LSTMs capture temporal dependencies and long-range patterns. Hybrid architectures, such as DeepConvLSTM and AttSense, leverage both strengths, often outperforming pure LSTM networks [349, 350]. This combination allows the system to model the full complexity of human movements, integrating both instantaneous signal patterns and sequential dynamics. For experimental evaluation, specific configurations of CNN, LSTM, and CNN-LSTM models were defined for both the Baseline and LAT implementations, providing a consistent framework to quantify the performance benefits of the trigger-based design.

For the Baseline classifier, where the primary objective is high accuracy, the following deep network topologies considered: (i) an architecture consisting of three stacked LSTM layers followed by three dense layers (3LSTM-3D), (ii) a CNN composed of three convolutional layers followed by three dense layers (3Conv-3D), (iii) an hybrid network featuring two convolutional layers and two LSTM layers, followed by three dense layers (2Conv-2LSTM-3D). To mitigate overfitting, a Dropout layer with a probability of 0.2 is inserted after each dense layer. For the LAT, the design is strictly constrained by energy efficiency ($E_{BASE} \gg E_{LAT}$). To meet this requirement, the depth and complexity of the baseline networks are reduced to create lightweight binary classifiers: (i) a simplified network comprising two LSTM layers and a single dense layer (2LSTM-1D), (ii) a compact CNN with two convolutional layers and a single dense layer (2Conv-1D), and (iii) a reduced hybrid model with one convolutional layer and one LSTM layer, followed by a single dense layer (1Conv-1LSTM-1D). This configuration facilitates a direct analysis of the trade-off between the detection capability of the trigger and its computational cost.

5.2.2 Semantic Template via Siamese Neural Networks

After the hierarchical trigger has successfully isolated a window of relevant motion, the system faces the core challenge of identification. Traditional HAR pipelines typically treat this phase as a static multi-class classification problem. However, the operational reality of facility management is inherently dynamic. Cleaning protocols evolve, and staff may

perform necessary tasks that were not captured during the initial data collection. Relying on a rigid classifier would require continuous, expensive retraining cycles to accommodate these variations. To overcome this limitation, it is necessary to make a paradigm shift from categorical classification to template-based semantic matching. In this framework, the objective is not to memorize specific classes, but to learn a robust metric of similarity. By leveraging Siamese Neural Networks to generate synthetic prototypes that distill the essential kinematic features of an activity into a reference vector.

The Main Network

The activity recognition framework is structured using a One-Vs-All (OVA) decomposition, in which each activity class is evaluated independently against all other classes through a binary classification scheme. Each binary classifier is implemented as a Siamese neural network, which computes the similarity between an unknown activity sample and a set of reference templates corresponding to the predefined classes. During inference, the input sample is compared to all templates, and the class associated with the template yielding the highest similarity score is assigned as the predicted activity.

Figure 5.5 illustrates the Siamese network employed in the three operational phases, namely *training*, *template generation*, and *activity recognition*. The core architecture, shown in Figure 5.5(a), is designed to process pairs of input samples and assess their degree of similarity. To this end, the network is composed of two identical branches, each implemented as a two-layer CNN. The two branches share the same weights and filters, ensuring that both inputs are mapped into a common feature space. Each branch inde-

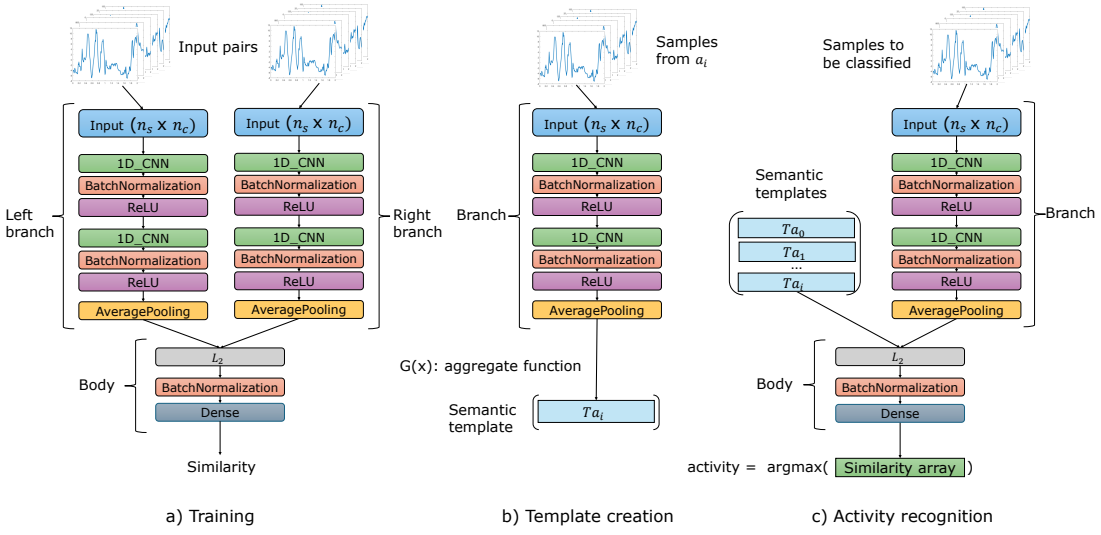


Figure 5.5: Structure of the network during three different operational phases: training (a), template creation (b), and activity recognition (c).

pendently extracts a latent representation from one of the two activity samples forming the pair. These feature vectors are then compared by computing their Euclidean distance,

expressed as the L_2 norm between the left and right embeddings. Formally, given the feature vectors L and R produced by the two branches, the distance is defined as:

$$d(L, R) = \sqrt{\sum_{i=1}^{N}(L_i - R_i)^2} \tag{5.17}$$

The resulting distance vector is subsequently processed through a Batch Normalization layer and a Dense layer. The final output consists of a single neuron with a logistic activation function, producing a scalar value in the range $[0, 1]$, where values close to 1 indicate that the two input samples belong to the same activity, while values close to 0 denote dissimilar activities.

During training, the Siamese network is fed with pairs of samples instead of individual instances. Two types of pairs are generated: positive pairs, consisting of samples from the same activity class, and negative pairs, consisting of samples from different classes. Formally, given an original HAR dataset of individual samples $[n_s \times n_c]_i$ with labels $l_i \in \mathcal{C}$, the training data is transformed into pairs $([n_s \times n_c]_L, [n_s \times n_c]_R)_j$ with binary labels $l'_j \in 0, 1$. The label assignment follows the rule: $l'_j = 1$ if both samples share the same class, and $l'_j = 0$ otherwise.

To ensure the network learns a similarity function representative of the full activity space, the training set includes positive pairs for all classes in $\mathcal{C}$ and negative pairs covering all class combinations. The network is trained using binary cross-entropy loss, encouraging high similarity scores for positive pairs and low scores for negative pairs. Instead of performing direct activity classification, the Siamese network learns a semantic embedding of the sensor data. This embedding captures the intrinsic characteristics of each activity and is subsequently used to generate class templates, which serve as reference points for comparing unknown samples during inference.

Semantic Template Generation

Automatically deriving a representative template for an activity class is a challenging problem in sensor-based HAR. Each activity sample is described by multiple time series acquired from heterogeneous wearable sensors, resulting in high-dimensional inputs with rich temporal dynamics. Identifying a compact representation that captures the essential characteristics of an activity is therefore non-trivial. Moreover, an effective template must be sufficiently distinctive to avoid overlap with templates of other classes, while remaining flexible enough to capture the natural intra-class variability caused by different users, execution styles, or contextual conditions.

Rather than generating synthetic templates directly from raw sensor signals, we exploit the feature extraction capability of the Siamese network to construct a semantic representation of each activity. As illustrated in Figure 5.5(b), the Siamese architecture is partitioned by isolating a single branch, and the output of the Average Pooling layer is used as the feature embedding. A set of samples belonging to the i-th activity class is processed through this branch, producing a collection of feature vectors denoted as F_i. The semantic template associated with the i-th class, indicated as Tc_i, is then obtained

by aggregating these feature vectors through a suitable function G, according to

$$Tc_i = G(F_i), \quad i \in \mathcal{C} \tag{5.18}$$

The aggregation step is essential to ensure that the resulting template is representative of the entire class rather than a single instance.

The simplest approach relies on the *arithmetic average* (AA), which computes the mean of all feature vectors extracted from the samples of a given class:

$$AA_i = \frac{1}{M}\sum_{j=1}^{M} F_{i,j} \qquad , \; i \in \mathcal{C} \tag{5.19}$$

While this method captures the central tendency of the class, it treats all samples equally, regardless of their representativeness. To address this limitation, a *similarity weighted average* (SWA) is introduced, where each feature vector is weighted by its average similarity score with respect to other samples of the same class:

$$SWA_i = \frac{\sum_{j=1}^{M} \overline{S_{i,j}} F_{i,j}}{\sum_{j=1}^{M} \overline{S_{i,j}}} \quad , \; i \in \mathcal{C} \tag{5.20}$$

Here, $F_{i,j}$ denotes the feature vector of the j-th sample belonging to class i, while $\overline{S_{i,j}}$ represents the average similarity value computed by comparing that sample with a randomly selected subset of samples from the same class. This formulation assigns stronger influence to samples that are more consistent with the overall class structure. A further refinement is provided by the *similarity conditioned average* (SCA), which explicitly filters out samples whose similarity falls below a class-dependent threshold Sth_i:

$$SCA_i = \frac{1}{P}\sum_{j=1}^{P} F_{i,j} | \overline{S_{i,j}} \geqq Sth_i, where \; P \leqq M, i \in \mathcal{C} \tag{5.21}$$

In this case, only the subset of P samples that satisfy the similarity constraint contributes to the template construction. The rationale behind both similarity-based strategies is that samples exhibiting low similarity with other instances of the same class are likely to be noisy, poorly executed, or atypical, and should therefore have a reduced impact on the final template or be excluded altogether. Including such samples risks producing templates with limited discriminative power. At the same time, excessive filtering may lead to overly rigid templates that fail to generalize to legitimate variations of the activity, improving precision at the expense of recall. The aggregation strategy thus plays a critical role in balancing robustness and generalization, directly influencing the effectiveness of the subsequent activity recognition phase.

Activity Recognition

Once the class templates have been created, the system can recognize unknown activities, as illustrated in Figure 5.5(c). In this phase, the Siamese network is reconfigured by removing one branch and by directly feeding the class templates into the Euclidean distance layer (L_2). Each unknown sample is first processed by the remaining branch to extract its feature representation. This feature vector is then compared with all class templates, producing a similarity score for each class. The sample is finally assigned to the class that yields the highest similarity value, using an *argmax* operation.

Compared to a traditional multi-class classifier, the proposed approach has lower computational complexity. This is mainly due to the lightweight structure of the Siamese network, which makes it suitable for deployment on resource-constrained devices. It should be noted, however, that the comparison between the extracted features of the unknown sample and the templates must be repeated for each class. This operation can be efficiently optimized by using parallel or multithreaded implementations.

A key advantage of the proposed method is its ability to support GZSAR. After training, the Siamese network learns a semantic feature space that captures the essential characteristics of activities. Thanks to this property, meaningful features can also be extracted from samples belonging to previously unseen classes ($\mathcal{U}$). By generating templates for these unseen classes, they can be included in the recognition process without retraining the network. In particular, during training, the system observes a set of labeled samples $S_{train} \in \mathcal{S}$, while during testing it operates on a set $S_{test} \in \mathcal{C} = \mathcal{S} \cup \mathcal{U}$. However, due to the *seen-class-bias* of the GZSAR problem, the samples from unseen classes tend to be incorrectly classified as belonging to seen classes. To reduce this effect, we adopt a modified version of the calibrated stacking mechanism proposed by Changpinyo et al. [325]. Specifically, we compute a calibrated similarity $\hat{S}_{i,j}$ between the j-th sample s_j and the template Tc_i of the i-th class using the following equation:

$$\hat{S}_{i,j} = S_{i,j} - \alpha \cdot \theta(s_j) \tag{5.22}$$

where $\theta(s_j)$ is equal to 1 if the sample belongs to a seen class and 0 otherwise. With this formulation, the similarity scores of seen classes are reduced by a factor α, while those of unseen classes remain unchanged. The parameter α, referred to as the *template calibration factor*, controls the strength of this correction and helps balance the competition between seen and unseen classes.

Finally, this framework naturally supports class-incremental learning. Adding a new activity does not require retraining the model; it only requires the creation of a new template for the corresponding class. This property is particularly important for constrained devices, which may not be able to handle the computational cost of retraining. In a practical scenario, a wearable HAR device could simply download new templates from the cloud to extend its recognition capabilities, while keeping the on-device computation lightweight.

Network configuration

After defining the activity representation strategy, the template generation process, and the recognition phase, we describe the architecture and configuration of the Siamese network used in the experimental evaluation. Artificial neural networks are complex systems whose computational performance depends on several factors, including architectural choices, hyperparameters, and the number of trainable parameters. Since the quality of the extracted semantic features directly affects both template creation and activity recognition, careful network configuration is required. To identify an effective set of hyperparameters for the Siamese network, an automated tuning process was adopted. The search was performed using the Hyperband Tuner [351], an efficient hyperparameter optimization algorithm that evaluates multiple configurations by training them for a limited number of epochs, progressively discarding poorly performing candidates. This procedure allows the exploration of a large hyperparameter space while keeping the computational cost manageable.

The tuning process produced an optimal configuration, denoted as *model #0*. From this base, two additional variants, *model #1* and *model #2*, were derived by progressively reducing the number of filters and kernel sizes, aiming to decrease both computational complexity and memory footprint. This approach targets deployment on resource-constrained embedded devices. All models were trained for 100 epochs using the Adam optimizer with a learning rate of 0.001. Sensor signals were segmented into sliding windows of 2 seconds (100 samples per window) with 75% overlap, providing a balance between temporal resolution and robustness of extracted features. The final hyperparameters for the three configurations are summarized in Table 5.1. Model #0 corresponds to the most expressive architecture, while models #1 and #2 offer progressively lighter alternatives suitable for low-resource scenarios.

model	Filter 1	Filter 2	Kernel 1	Kernel 2
#0	62	54	11	4
#1	50	35	8	4
#2	30	26	5	3

Table 5.1: Values for the hyperparameter of the three Siamese Network models (#0, #1, #2).

5.2.3 Tiny-HAR Transformer

While the previously discussed Siamese framework addresses the issue of adaptability to new activity classes, ensuring the robust interpretation of complex, long-duration protocols requires a fundamentally different architectural perspective. Traditional deep learning models, such as CNNs, RNNs, and LSTMs, have long served as the standard for HAR. However, they are restricted by their local receptive fields, capturing only short-term patterns, and often struggle to maintain context over long sequences due to memory constraints and vanishing gradient issues. This deficiency is particularly detrimental in the context of sanitization workflows, where a single activity (i.e., vacuuming a corridor) is not merely

a repetitive gesture but a complex temporal sequence where the beginning and end are semantically linked across a significant time window. Transformers address this by discarding sequential processing in favor of a global view, utilizing self-attention mechanisms to correlate disparate signal segments regardless of their temporal distance. However, the theoretical advantages of these large models must be weighed against the challenges of deploying such resource-intensive architectures on the energy-constrained embedded devices that form the backbone of sensor-based HAR systems.

To evaluate the feasibility of Transformer-based models on wearable and resource-constrained devices, the analysis starts from a lightweight Transformer configuration proposed by Dirgova et al. [338]. Building upon this baseline, hyperparameter fine-tuning and model compression techniques are applied to reduce computational and memory requirements. Model performance is characterized in terms of classification accuracy and inference latency using a real-world HAR case study deployed on an Espressif ESP32, which represents a typical low-power embedded platform (described in Section 3.3.5). The adopted evaluation methodology follows the same experimental protocol previously applied to CNN and LSTM models, enabling a direct comparison between Transformer-based architectures and more traditional deep learning approaches [352].

Figure 5.6 illustrates a minimal Transformer configuration. The architecture consists of a positional encoding layer, a single encoder block, and a final multilayer perceptron composed of two fully connected layers. The encoder block includes a normalization layer and a multi-head attention layer, while dropout layers are applied both within the encoder and the final classifier to reduce overfitting.

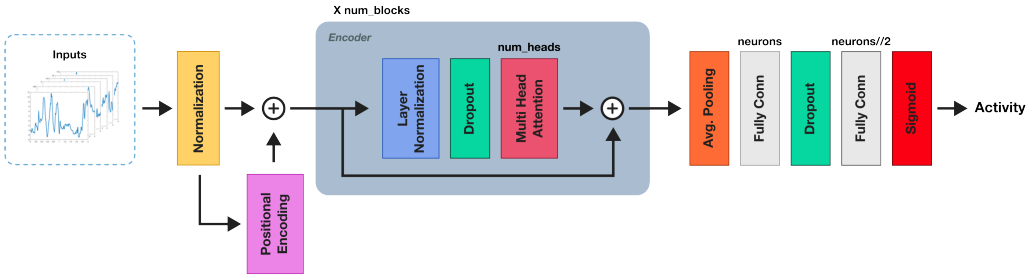


Figure 5.6: Schema representing a minimal Transformer configuration for sensor-based HAR

Model dimensionality plays a critical role when targeting tiny devices. In particular, embedded deployment requires satisfying strict memory and execution time constraints. During inference, model weights and instructions are accessed in a read-only manner. On microcontroller units (MCUs) such as the ESP32, these elements can be stored directly in non-volatile memory (ROM or flash), which is typically larger than available RAM. This allows the model to be deployed alongside the firmware without excessive memory overhead. In contrast, input data acquired from onboard sensors must be stored in RAM during inference. In sensor-based HAR applications, sensor signals are segmented into short time windows lasting a few seconds, resulting in memory footprints of a few hundred

kilobytes, which are well within the RAM constraints of modern MCUs. From a temporal perspective, inference must be completed within the duration of each sampling window to guarantee continuous real-time data acquisition and processing.

The minimal architecture, referred to as a tiny HAR-Transformer, can be scaled by increasing three main structural parameters: (i) the number of encoder blocks (`num_blocks`), (ii) the number of attention heads per block (`num_heads`), and (iii) the number of neurons in the final multilayer perceptron (`neurons`). Each encoder block introduces additional residual connections, while each attention head internally includes its own residual structure. After an initial manual exploration of the architectural space, the largest Transformer configuration satisfying both memory and execution time constraints on the target device was identified. Starting from this upper-bound configuration, the model complexity was progressively reduced by decreasing the aforementioned parameters.

Each Transformer configuration was trained and evaluated on three publicly available HAR datasets to assess classification performance. The trained models were then compressed, converted to C code, and deployed on the ESP32 development board. During on-device execution, inference latency was measured to evaluate real-time feasibility. This workflow provides a systematic assessment of the trade-off between recognition accuracy and computational efficiency, offering insight into the practical applicability of Transformer-based models for wearable HAR scenarios.

5.2.4 Large Language Models in HAR

The success of Transformer-based models in Natural Language Processing (NLP) demonstrates their remarkable ability to capture long-range and complex dependencies in sequential data. This insight motivates a paradigm shift in sensor-based HAR, where human motion is treated not as a simple signal processing task, but as a linguistic modeling problem. By adapting architectures such as BERT (Bidirectional Encoder Representations from Transformers), inertial sensor streams can be interpreted as sequences of discrete movement tokens forming coherent activity sentences. This approach exploits masked language modeling to handle missing or corrupted segments, enabling reconstruction of incomplete sensor streams. Additionally, it facilitates the generation of realistic synthetic activity traces, addressing the chronic scarcity of labeled datasets commonly encountered in hygiene management and wearable HAR applications.

Signal Tokenization

Tokenization represents a critical step in the preprocessing pipeline for training LLMs on time-series data, such as those generated in sensor-based HAR. Unlike natural language, where tokens correspond to words or subword units with semantic meaning, time-series data consist of continuous numerical values sampled at regular intervals. Consequently, the tokenization procedure must capture temporal patterns while preserving inherent dependencies in the data. This tokenization process consists of two main stages: (i)segment encoding and (ii) symbolic representation. During segment encoding, accelerometer signals are divided into chunks of fixed length. Each chunk is then passed through a pre-trained

autoencoder that compresses it into a low-dimensional embedding. This embedding is subsequently transformed into a symbolic token using a k-means clustering algorithm.

Figure 5.7 illustrates the overall tokenization and detokenization procedure. A window of eight consecutive samples is processed by the encoder, which compresses the input by halving its dimensionality. The encoder architecture consists of two convolutional layers

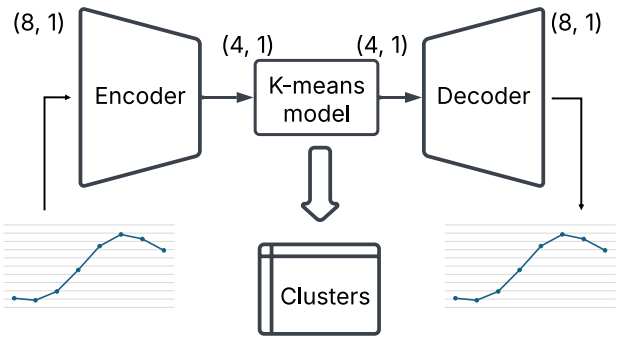


Figure 5.7: The main tokenization procedure of a time series using the Convolutional Autoencoder model.

followed by two dense layers, with the last dense layer representing the latent space. The decoder reconstructs the original signal from the latent embedding using a dense layer, followed by three transposed convolutional layers and a final dense layer. Linear activations are applied in all layers except for the two dense layers preceding the latent space, which use ReLU activations. Architectural details of the encoder and decoder are reported in Table 5.2. Once the latent embedding is obtained, a k-means clustering module gener-

Table 5.2: Hyperparameters of the encoder/decoder architecture used in segment encoding.

Encoder		Decoder	
Layer type	Output shape	Layer type	Output shape
input	(8, 1)	dense	(5)
conv1d	(8, 32)	repeat_vector	(8, 5)
conv1d	(8, 16)	conv1d_transpose	(8, 8)
flatten	(128)	conv1d_transpose	(8, 16)
dense	(5)	conv1d_transpose	(8, 32)
dense	(4)	dense	(8, 1)

ates the final token by assigning each embedding to the closest centroid. The vocabulary size of tokens is determined by the number of centroids specified during k-means training. Selecting an appropriate vocabulary size is essential, as it determines the trade-off between representational expressiveness and model efficiency. A larger vocabulary reduces quantization error but increases model complexity.

To evaluate this trade-off, multiple autoencoder/k-means configurations were trained on the full dataset, and the tokenization/detokenization performance was measured in

terms of Mean Absolute Error (MAE) relative to the original signal. Table 5.3 reports the MAE for different vocabulary sizes. The results indicate that increasing the number of tokens reduces the reconstruction error, with diminishing returns beyond 2,000 centroids. For this reason, a vocabulary of 2,000 tokens was adopted in the current methodology, balancing accuracy and computational cost.

Table 5.3: Performance of the tokenization/detokenization when varying the size of the vocabulary.

Vocabulary size	MAE
500	0.060
1,000	0.050
1,500	0.045
1,600	0.044
2,000	0.042
2,500	0.039
3,000	0.038
$4 * 2^{32}$	0.020

To further characterize the tokenization procedure, the coordinates of the k-means centroids, mapped back into the accelerometer signal domain, are visualized in the scatter plot reported in Figure 5.8. Since each centroid is defined in an eight-dimensional space, a dimensionality reduction is applied using the t-distributed Stochastic Neighbor Embedding (t-SNE) technique to obtain a two-dimensional representation suitable for visualization. The resulting distribution of centroids is clearly non-uniform and reveals the presence of multiple super-clusters in which several tokens occupy nearby positions in the accelerometer space. This structure indicates that groups of tokens encode highly similar signal patterns. As a consequence, within certain bounds, substituting a token with another belonging to the same or a nearby region is expected to produce only minor variations in the reconstructed signal, still allowing for a satisfactory approximation of the original time series.

Self-Supervised training of LLMs

To harness the capabilities of LLMs for HAR, we adopted a standard Bidirectional Encoder Representations from Transformers (BERT) architecture initialized from scratch and equipped with a custom vocabulary of 2,000 tokens. The training data consisted of continuous accelerometer streams segmented into fixed-size windows of 256 samples, corresponding to a temporal duration of approximately 5.12 seconds at 50 Hz. Through the tokenization process, each window was compressed into a sequence of 32 discrete tokens. The dataset was subsequently split into training (80%) and testing (20%) sets.

To enable the model to learn robust temporal dependencies and latent representations of human movements, we implemented a Masked-Language Modeling (MLM) self-supervised training procedure. Within each sequence of 32 tokens, a random subset ranging from

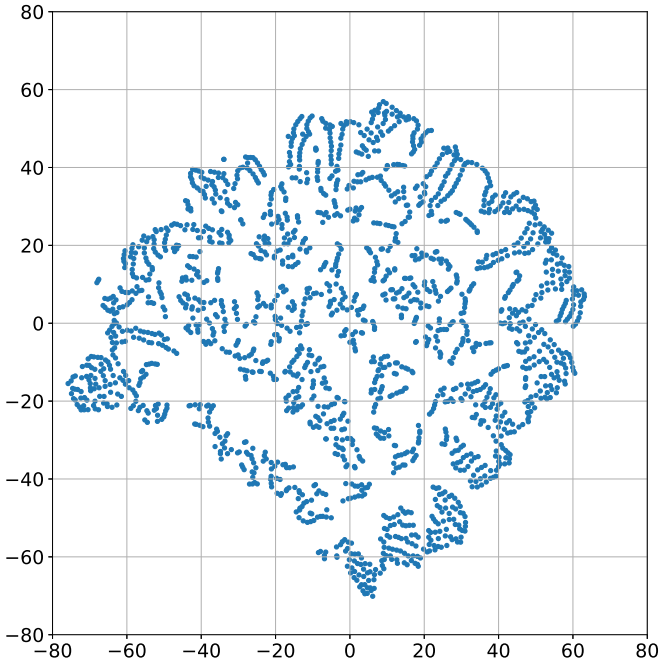


Figure 5.8: Scatter plots of the coordinates of 2,000 k-means centroids trained on top of the entire dataset. The dimensionality reduction of the coordinate has been obtained using t-SNE.

1 to 12 tokens was replaced with a special $[MASK]$ token. The model was then optimized to reconstruct these missing tokens based on the bidirectional context provided by the surrounding unmasked data. To further enhance the model's ability to capture both short-term and long-term dependencies, we applied two distinct masking strategies during training. The first, Random Token Masking, involves masking independent tokens at random positions. This encourages the model to learn general feature representations. The second, Span Masking, involves masking consecutive segments of tokens. This forces the model to understand the underlying temporal structure over longer horizons, as it cannot rely on immediate neighbors for reconstruction.

Once pre-trained, the BERT model was repurposed to generate synthetic sensor data. Analogous to textual prompting in NLP, we utilized embeddings of real-world signal chunks as "prompts" to condition the model. By providing a specific context of real motion data, we induce the model to generate new traces that statistically resemble a specific human activity. We evaluated two distinct inference strategies for this generative process: (i) multi-step generation, and (ii) single-step generation. Multi-step generation involves predicting n tokens simultaneously in a single inference pass. The size of the context provided to BERT is adjusted dynamically depending on the number of tokens to be generated. Single-step generation, also called auto-regressive, leverages an iterative approach. Starting with a context of 31 tokens ($\approx$ 5 seconds of real signal), the model predicts the 32^{nd}

token. Subsequently, the context window is shifted by one position, and the newly generated token is appended to the end to serve as context for the next step. This procedure is repeated n times. A key characteristic of this strategy is that as the iteration progresses, the context transitions from real data to fully synthetic data. After 31 iterations, the model generates new samples based entirely on its own previous predictions, effectively hallucinating plausible continuations of the activity.

Performance Assessment

Assessing the effectiveness of an LLM in processing time-series data requires a multi-faceted evaluation strategy. Since the model operates in a discrete token space but represents continuous physical signals, distinct metrics must be adopted to evaluate both the reconstruction accuracy and the semantic validity of the generated data. To quantify the model's ability to fill in missing chunks of data, we measured both the token prediction accuracy and the reconstruction error. These two metrics offer complementary perspectives on the model's performance. The token prediction accuracy serves as a direct measure of the model's performance in the discrete domain. It is calculated as the percentage of masked tokens that are correctly predicted by the LLM. However, this metric tends to be too conservative. It treats all misclassifications equally, failing to account for the semantic proximity of different tokens. In the context of our quantized dictionary, an incorrect token prediction might still correspond to a signal chunk that is mathematically very similar to the ground truth. To address this, we also computed the reconstruction error in the signal domain. This involves decoding the predicted tokens back into time-series values and calculating metrics such as the Mean Squared Error (MSE) or Mean Absolute Error (MAE) against the ground truth signal. Comparing these two metrics reveals that a low token accuracy does not necessarily imply poor signal reconstruction. If the model predicts a wrong token that is nonetheless adjacent to the correct one in the code-book space, the resulting signal error may be negligible. Therefore, while token accuracy measures the model's classification precision, the reconstruction error provides a truer estimate of the physical fidelity of the imputed data.

Beyond numerical reconstruction errors, it is critical to verify that the synthetically generated signals semantically correspond to plausible human activities. To this end, we employed a classification-as-a-metric approach. We utilized a state-of-the-art HAR classification model, specifically the architecture presented by Bigelli et al. [353], which consists of four convolutional layers followed by average pooling and four dense layers. The underlying assumption is that if the generated data preserves the distinct features of the target activity, such as the rhythmic pattern of walking or the intensity of running, the classifier should be able to correctly infer the activity class from the synthetic signal alone. We assume that if the inferred class matches the label of the context prompt used to generate the signal, the generation is considered a faithful representation of that human activity. The quality of the generated traces was evaluated across two distinct benchmarks to test different capabilities of the system. In the first experiment, we used data from the training dataset as the input context. This tests the model's ability to reproduce and complete patterns it has already seen during the pre-training phase. In the

second set of experiments, we forced the system to generate data based on contexts from a completely unknown dataset containing unseen human activities. This latter experiment is particularly significant since success in this benchmark would demonstrate that the model has not merely memorized specific dataset patterns but has effectively learned the general "language of the accelerometer" regardless of the specific activity being performed.

5.3 Experimental Setup

5.3.1 Sensor-Based HAR Datasets

To evaluate the robustness and generalization of the proposed methodologies, experiments were conducted on a diverse collection of publicly available HAR datasets. The selection criteria were designed to cover a wide range of sensing modalities, device types, and body placements, as well as activities of varying complexity, from basic locomotion patterns to fine-grained, hand-centric daily tasks. This diversity ensures that the results reflect the performance of the methods under realistic, heterogeneous operational conditions.

The **MotionSense** dataset contains tri-axial accelerometer and gyroscope data sampled at 50 Hz and collected using an iPhone 6s positioned in the front pocket of the participants [354]. It includes six activities (walking, jogging, walking upstairs, walking downstairs, sitting, and standing) performed by 24 subjects with heterogeneous demographic characteristics in terms of age, height, weight, and gender.

The **MobiAct** dataset [355] contains accelerometer and gyroscope recordings sampled at 50 Hz from 66 participants performing 12 activities. These activities span static postures, dynamic movements, and transitional actions, such as standing, walking, jogging, jumping, and sit-to-stand transitions, providing a comprehensive testbed for assessing model robustness across varied motion dynamics.

The **RealWorld** dataset [356] comprises movement data from 15 subjects (8 male, 7 female) performing a broad range of activities, including standing, sitting, lying, walking, running, jumping, and stair climbing. A unique feature of this dataset is the simultaneous acquisition from seven body locations (chest, forearm, head, shin, thigh, upper arm, and waist) and multiple sensor modalities, including accelerometer, gyroscope, magnetic field, GPS, light, and sound. This diversity allows for evaluating models under multi-sensor and multi-placement scenarios, reflecting realistic deployment conditions.

The **WISDM** Smartphone and Smartwatch Activity and Biometrics Dataset [357] includes recordings from 51 participants, each performing 18 activities for approximately three minutes per task. The activities range from basic locomotion to fine-grained hand-oriented actions, such as typing, folding clothes, brushing teeth, and eating. Data were sampled at 20 Hz using both a smartwatch worn on the dominant wrist and a smartphone carried in the front pocket, allowing the analysis of cross-device and cross-placement sensing.

The **UCI HAR** dataset [358] is a widely used benchmark in HAR research. It contains accelerometer and gyroscope recordings sampled at 50 Hz from a smartphone worn at the waist by 30 subjects aged 18–48 years. The dataset covers six core activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying, offering a standard

reference for evaluating fundamental motion recognition tasks.

Finally, the **Watch HAR** dataset [359] focuses on wrist-worn sensing and was collected in a controlled laboratory environment. It comprises data from 13 users performing everyday domestic activities, such as reading, typing, washing dishes, washing hands and face, and using a remote control, in addition to standard locomotion tasks. The dataset includes accelerometer, gyroscope, and magnetometer measurements, providing a challenging benchmark for recognizing complex and subtle hand movements.

Together, these datasets enable a comprehensive evaluation of the proposed approaches across heterogeneous sensing configurations and activity domains.

5.3.2 Hardware Platforms

The setup encompasses a diverse range of computing platforms, ranging from high-performance workstations used for the training phases to resource-constrained edge devices adopted for inference. The experiments and validations regarding wearable devices were conducted on an OPPO Watch 46mm. This device runs on the Qualcomm Snapdragon Wear 3100 platform, featuring 1 GB of RAM and 8 GB of non-volatile flash memory. The smartwatch integrates a comprehensive sensor suite with accelerometer, gyroscope, magnetometer, and barometer, and operates on Wear OS, a version of Android made especially for wearable devices. The operating system supports the execution of TensorFlow Lite models, enabling on-device inference. For the purpose of electrical characterization, the device was disassembled, and the internal 430 mAh Li-Po battery was bypassed to connect the power rails directly to the measurement setup. To extend the analysis to highly constrained hardware, the Espressif ESP32 microcontroller was employed. A detailed description of these devices can be found in Section 3.3.5.

The experimental workload was distributed across two distinct high-performance computing environments, allocated based on the specific computational requirements of the models being developed. The training and validation of the Lightweight Accurate Trigger (LAT) techniques, as well as the experiments concerning the semantic template generation, were conducted on a workstation equipped with two Intel® Xeon® Silver 4314 CPUs and three NVIDIA® A100 GPUs. This configuration provided the necessary memory bandwidth and tensor core performance required for the optimization of the trigger mechanisms and the semantic feature extractors. Conversely, the computationally intensive tasks related to the self-supervised pre-training of the BERT architecture and the extensive exploration of the hyperparameter space were performed on a separate, higher-density infrastructure. This workstation features two Intel® Xeon® Gold 6426Y processors, 500 GB of system RAM, and eight NVIDIA® L40S GPUs. On this specific hardware configuration, the complete self-supervised training procedure for the LLM requires approximately eight days of continuous computation, while the subsequent inference phase for data generation operates at approximately 3 ms per sample.

5.3.3 Classification Metrics

To quantitatively assess the performance of the models, we adopted specific evaluation protocols depending on the nature of the learning task. For standard supervised learning

scenarios, we employed conventional macro-averaged metrics. Conversely, for the Generalized Zero-Shot Activity Recognition (GZSAR) experiments, we utilized a specialized set of metrics designed to measure the balance between seen and unseen class recognition.

For the general multi-class classification problems, performance was evaluated based on the elements of the confusion matrix calculated for each class i (where $i \in [1 \dots N]$):

- TP_i: True Positives for class i.
- TN_i: True Negatives for class i.
- FP_i: False Positives for class i.
- FN_i: False Negatives for class i.

To synthesize these individual values into global performance indicators, particularly in the presence of potential class imbalances, we adopt the macro-averaging approach [360]. This method involves calculating each metric for every class independently and then computing the arithmetic mean. This ensures that every human activity class contributes equally to the final score, preventing dominant classes from skewing the results.

The most immediate measure of system performance is **Accuracy**, which represents the overall ratio of correctly classified samples relative to the total number of samples:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{5.23}$$

However, accuracy alone often fails to capture the nuances of error types. To address this, we examine **Precision**, which measures the reliability of the model's positive predictions, effectively answering the question: "Of all the instances labeled as class i, how many were actually class i?"

$$Precision = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i} \tag{5.24}$$

Complementing precision is **Recall**, which assesses the model's coverage capability. It answers: "Of all the instances that actually belong to class i, how many did the model successfully find?"

$$Recall = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i} \tag{5.25}$$

Finally, since there is often a trade-off between Precision and Recall, we utilize the **F1-Score**. This metric serves as the harmonic mean of the two, providing a single, balanced value that penalizes extreme disparities, thus offering a more robust assessment of the classifier's effectiveness:

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.26}$$

In the context of Generalized Zero-Shot Activity Recognition (GZSAR), the evaluation is more complex because the test set contains samples from both seen classes ($\mathcal{S}$, observed

during training) and unseen classes ($\mathcal{U}$, never observed during training). Evaluating performance on the combined set $\mathcal{C} = \mathcal{S} \cup \mathcal{U}$ requires monitoring whether the model is biased toward the classes it has already seen. Following the protocol established by Wang et al. [361], we first compute the average per-class accuracy as:

$$A = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{1}{N_{i,s}} \sum_{j=1}^{N_{i,s}} \mathbb{1}[y_pred_{i,j} == y_true_{i,j}] \tag{5.27}$$

where N_c is the number of classes in the target set ($\mathcal{S}$ or $\mathcal{U}$), and $N_{i,s}$ is the number of samples in the i-th class. We further refer to the accuracy of seen classes as $A_{\mathcal{S}\to\mathcal{C}}$, while for unseen classes $A_{\mathcal{U}\to\mathcal{C}}$. Notice that, for the sake of completion, in the experimental results section, we also reported the accuracy of seen classes obtained in a traditional multi-class experiment, formally $A_{\mathcal{S}\to\mathcal{S}}$, and the accuracy of the unseen classes in a conventional ZSAR experiment (i.e., without seen classes in the test phase), formally $A_{\mathcal{U}\to\mathcal{U}}$.

Finally, to provide a holistic view of the GZSAR performance, we compute the Harmonic Mean (HM) of the seen and unseen accuracies:

$$HM = \frac{2 \times A_{\mathcal{S}\to\mathcal{C}} \times A_{\mathcal{U}\to\mathcal{C}}}{A_{\mathcal{S}\to\mathcal{C}} + A_{\mathcal{U}\to\mathcal{C}}} \tag{5.28}$$

The HM is the most critical metric in this context. Standard classifiers often exhibit a strong bias toward seen classes (where $A_{\mathcal{S}\to\mathcal{C}} \gg A_{\mathcal{U}\to\mathcal{C}}$). Because this metric is sensitive to low values, a high HM score indicates that the model has successfully mitigated this bias and performs well on both seen and unseen categories simultaneously.

5.4 Experimental Results

This section provides the experimental validation of the methodologies developed to enhance wearable-based activity monitoring within the sanitization context. The analysis is structured to evaluate how these frameworks perform across different operational dimensions, from real-time energy management to the recognition of novel tasks and the feasibility of advanced sequence modeling. The first part of the evaluation focuses on the operational efficiency and flexibility of the system. This begins with the validation of the LAT, quantifying its ability to maintain high detection fidelity while significantly reducing the computational burden on wearable hardware. Subsequently, the effectiveness of the semantic template framework is assessed through the lens of GZSAR. The second part presents a comprehensive study of Transformer-based architectures and their application for sensor data. This analysis characterizes their performance in reconstructive and generative tasks, specifically evaluating their accuracy in data imputation and the statistical realism of generated synthetic activity traces. Finally, the discussion addresses the critical trade-off between inferential power and hardware constraints by reporting the execution metrics of these models when deployed on real-world, resource-constrained embedded platforms.

5.4.1 Lightweight Accurate Trigger

The first phase of the evaluation focused on identifying the optimal architecture for the Base Classifier. We explored the hyperparameter space of three distinct neural network topologies: a pure LSTM network (3LSTM_3D), a pure Convolutional network (3Conv_3D), and a hybrid architecture (2Conv_2LSTM_3D). Table 5.4 details the resulting architectures, where H_n represents the number of internal filters (for Conv layers) or cell units (for LSTM layers), and D_n represents the neurons in the dense layers. The table also

Table 5.4: Hyperparameter settings, model size, inference time, and energy consumption for the evaluated base classifiers.

parameter	**3LSTM_3D**	**3Conv_3D**	**2Conv_2LSTM_3D**
H1	544	192	320
H2	512	864	1024
H3	896	928	1024
H4	-	-	608
D1	416	416	512
D2	576	224	192
D3	#classes	#classes	#classes
size [kB]	10,101	10,137	10,000
time [ms]	$1,987 \pm 96$	681 ± 23	$1,808 \pm 88$
energy [mJ]	306 ± 12	123 ± 5	252 ± 11

reports the deployment metrics measured on the target hardware. While all models have a comparable memory footprint (10 MB), the 3Conv_3D network demonstrates superior efficiency. Its inference time is approximately 681 ms, resulting in an energy consumption of just 123 mJ per inference. In contrast, the LSTM-based models require nearly two seconds for a single inference, consuming up to 306 mJ.

Table 5.5 summarizes the classification performance. The 3Conv_3D network consistently outperforms the recurrent architectures across all datasets. For instance, on the WISDM dataset, it achieves an accuracy of approximately 96%, compared to 89% and 91% for the LSTM and Hybrid models, respectively. Based on its dominant performance in both accuracy and energy efficiency, the 3Conv_3D network was selected as the reference Base Classifier for the remainder of the evaluation.

Accuracy and Energy Performance

After defining the baseline classifier, the focus shifted to the design and optimization of the Lightweight Accurate Trigger (LAT). The exploration involved varying the number of hidden units within the proposed 1D architectures across the three binarized datasets. Figure 5.9 shows how classification accuracy evolves with network complexity. As expected, larger networks generally yield higher accuracy up to a certain limit. Beyond this point, as observed for the 2LSTM_1D on WISDM, further complexity can degrade performance. Across all tests, the purely convolutional model (2Conv_1D) proved to be the most reliable,

Table 5.5: Classification performance comparison on the three reference datasets.

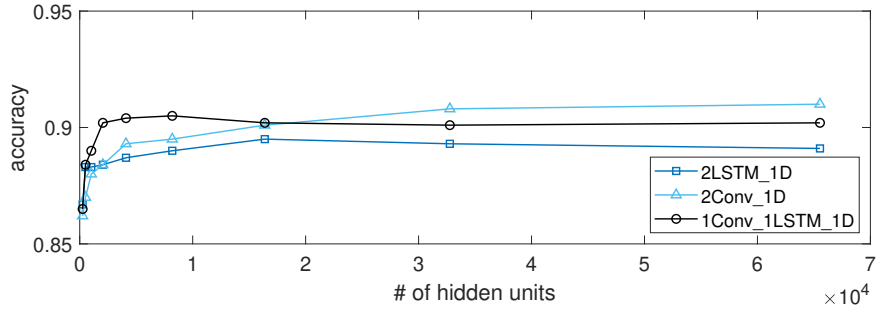
metric	dataset	3LSTM_3D	3Conv_3D	2Conv_2LSTM_3D
accuracy	Ad-hoc DB	0.878 ± 0.008	0.916 ± 0.004	0.886 ± 0.010
	Watch_HAR	0.821 ± 0.008	0.916 ± 0.009	0.824 ± 0.007
	WISDM	0.889 ± 0.006	0.958 ± 0.002	0.909 ± 0.008
precision	Ad-hoc DB	0.882 ± 0.006	0.915 ± 0.004	0.889 ± 0.003
	Watch_HAR	0.822 ± 0.004	0.908 ± 0.006	0.825 ± 0.007
	WISDM	0.887 ± 0.005	0.942 ± 0.005	0.909 ± 0.009
recall	Ad-hoc DB	0.881 ± 0.007	0.915 ± 0.003	0.894 ± 0.010
	Watch_HAR	0.817 ± 0.009	0.911 ± 0.014	0.823 ± 0.008
	WISDM	$0.887 \pm 0,007$	0.978 ± 0.003	0.907 ± 0.005
MCR	Ad-hoc DB	0.122 ± 0.008	0.084 ± 0.004	0.114 ± 0.010
	Watch_HAR	0.179 ± 0.008	0.084 ± 0.009	0.176 ± 0.007
	WISDM	0.111 ± 0.006	0.042 ± 0.002	0.091 ± 0.008

achieving accuracy comparable to or better than the hybrid 1Conv_1LSTM_1D models while maintaining lower inference latency.

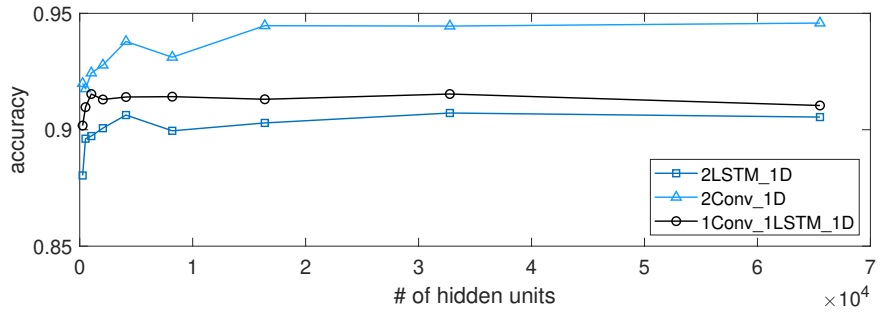
To select the optimal configuration for each dataset, we performed a Pareto analysis. Figure 5.10 plots the energy consumption of a single inference against the False Negative Rate (FNR) for each configuration. The optimal models, marked with red circles, represent the point that minimizes energy consumption while maintaining an acceptable error rate. On the ad-hoc database and the Watch HAR, the optimal balance is achieved by a 2Conv_1D network with 256 and 128 filters in the first and second layers, respectively. On the WISDM dataset, the optimal configuration is a 2Conv_1D network with 512 and 128 filters. Consequently, the 2Conv_1D architecture was selected as the standard deputy for the LAT system.

Table 5.6 presents the comprehensive performance of the complete triggering system (the cascaded LAT and the base classifier) using the optimal configurations identified above. The table reports both the theoretical misclassification rate (MCR_{system}), calculated via Equation 5.15, and the experimentally measured rate ($MMCR_{system}$), obtained by running the cascaded models on the test set. Theoretical calculations suggest a significant potential performance drop (i.e., on Ad-hoc DB, the error rate rises from 0.084 to 0.188). However, the measured overhead is substantially lower, ranging from just 0.2% to 1.1% ($\Delta MMCR$). This discrepancy occurs because the samples that are hard for the LAT (leading to false negatives) are largely the same samples that the base classifier would misclassify anyway. Consequently, the errors do not stack linearly. The LAT primarily rejects samples that the base classifier would likely have failed to recognize correctly, thereby preserving the overall system integrity.

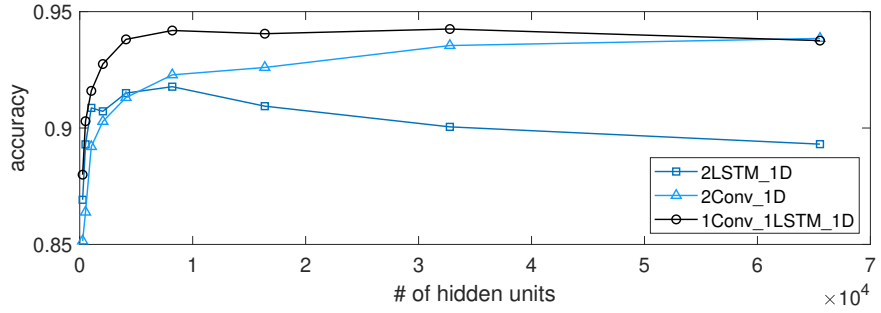
The primary motivation for the triggering approach is energy efficiency. As shown in Table 5.6, the LAT models are extremely lightweight (400 - 800 kB) compared to the base classifier (> 10 MB) and are significantly cheaper to run. The energy ratio between a



(a) Ad-hoc DB

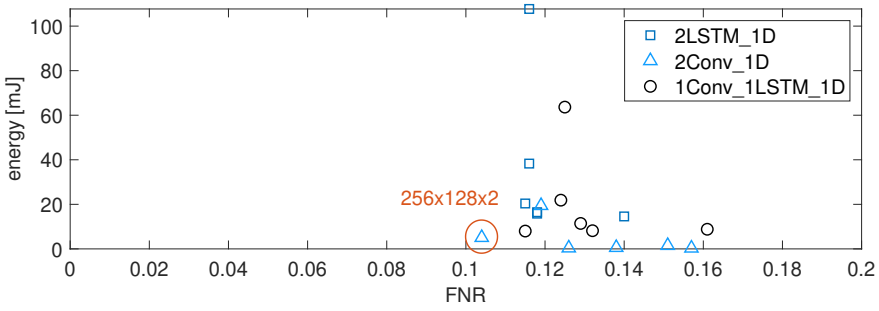


(b) Watch_HAR

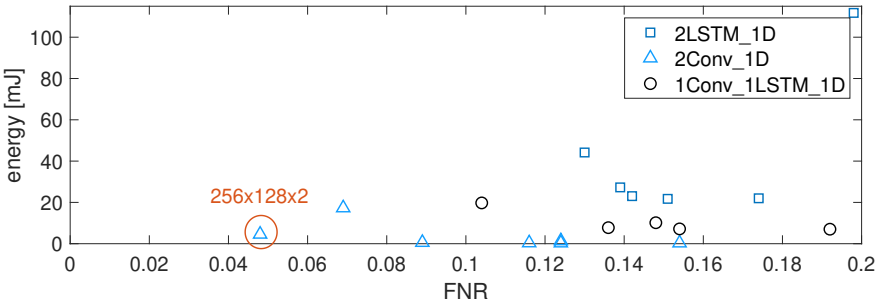


(c) WISDM

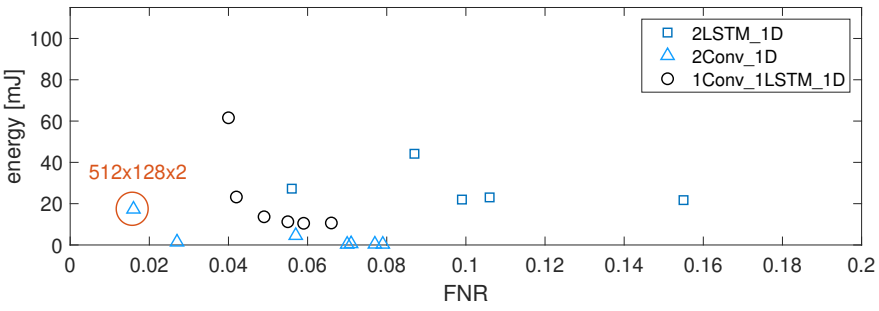
Figure 5.9: Accuracy obtained by the LAT models varying the total number of hidden units on top of the three datasets.



(a) Ad-hoc DB



(b) Watch_HAR



(c) WISDM

Figure 5.10: Pareto diagram reporting the FNR against energy consumed in a single inference by each LAT configuration.

Table 5.6: Best performance of the complete triggering system together with the model size, inference time, and energy consumption.

	Ad-hoc DB	**Watch_HAR**	**WISDM**
# of elements	256x128x2	256x128x2	512x128x2
FNR	0.104 ± 0.035	0.048 ± 0.011	0.016 ± 0.003
TPR	0.896 ± 0.035	0.952 ± 0.011	0.984 ± 0.003
FPR	0.102 ± 0.019	0.071 ± 0.014	0.027 ± 0.009
MCR_{base}	$\mathbf{0.084 \pm 0.004}$	$\mathbf{0.084 \pm 0.009}$	$\mathbf{0.042 \pm 0.002}$
MCR_{system}	0.188 ± 0.041	0.134 ± 0.029	0.059 ± 0.020
$MMCR_{system}$	$\mathbf{0.095 \pm 0.013}$	$\mathbf{0.091 \pm 0.009}$	$\mathbf{0.044 \pm 0.003}$
$\Delta MMCR_{system}$	$\mathbf{0.011 \pm 0.017}$	$\mathbf{0.007 \pm 0.018}$	$\mathbf{0.002 \pm 0.005}$
base classifier (3Conv_3D)			
size [kB]		10,137	
time [ms]		681 ± 23	
energy [mJ]		123 ± 5	
LAT			
size [kB]	423	423	808
time [ms]	58.38 ± 3.32	56.32 ± 5.61	213.60 ± 11.06
energy [mJ]	4.73 ± 0.94	4.57 ± 0.81	17.32 ± 6.30
E_{base}/E_{LAT}	$\mathbf{26.01 \pm 6.23}$	$\mathbf{25.89 \pm 5.87}$	$\mathbf{7.10 \pm 2.87}$

single inference of the base classifier and the LAT ranges from 7× to 26×, depending on the configuration. Figure 5.11 reports the energy savings achieved by the proposed approach as a function of $p(A)$, namely the probability that at least one target activity is present in the input signal. The energy consumption ECE is computed according to Equation 5.10, using the measured performance values summarized in Table 5.6. The results show that, for the WISDM dataset, when the probability of observing one of the target activities is $p(A) = 0.5$, the proposed solution yields an energy saving close to 40%. For the Watch HAR and Ad-hoc datasets, the achieved savings are even higher, exceeding 50% under the same conditions.

Even larger benefits can be expected in realistic deployment scenarios. Consider, for instance, a smartwatch-based application for monitoring hand washing and sanitization activities, such as the one described in [40]. Assuming that a user washes or sanitizes their hands 30 times per day, with an average duration of 30 seconds per event, the resulting activity probability is $p(A) = 30 \times 30/86400 \approx 0.01$. Under these conditions, the proposed approach can achieve an energy saving of approximately 95%. A similar reasoning can be applied to the WISDM dataset. If a user spends about 6 hours per day performing activities such as typing, brushing teeth, eating, drinking, writing, clapping, or folding clothes, the corresponding probability is $p(A) = 6/24 = 0.25$. In this scenario, the proposed architecture is able to reduce the overall energy consumption of the smartwatch by up to

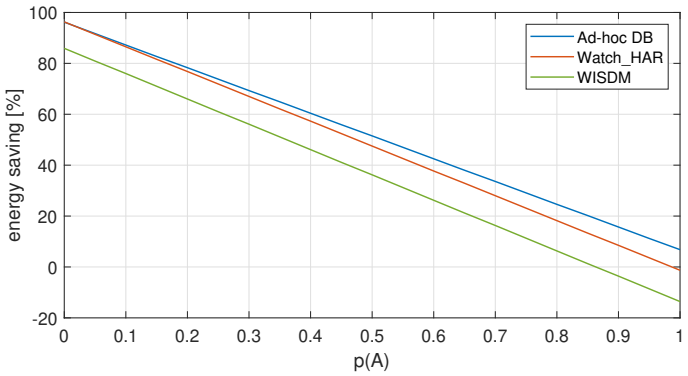


Figure 5.11: Theoretical energy saved by the triggering approach with respect to the baseline when varying the FNR of the LAT classifier for different values of the probability $p(A)$

60%, confirming its effectiveness in practical, activity-sparse monitoring applications.

Sensitivity to the Configuration Parameters

A potential concern with the proposed methodology is whether the performance of the LAT relies heavily on the specific semantic grouping of activities, such as distinguishing hand-oriented tasks from locomotion. To investigate this, we conducted a sensitivity analysis to determine how the choice of partitions A' and A'' affects system stability. We performed a series of experiments using the WISDM dataset, where activities were randomly assigned to the two partitions, rather than being grouped by semantic logic. This randomized workflow was repeated ten times using different random seeds, allowing us to measure both average performance and standard deviation under varying conditions.

Table 5.7 summarizes the results of this stress test. The results reveal a high degree

Table 5.7: Classification performance of the LAT and the base classifier under random activity partitioning for the WISDM dataset.

	accuracy	precision	recall
Classifier	0.9425 ± 0.0061	0.9413 ± 0.0062	0.9307 ± 0.0065
LAT	0.9559 ± 0.0115	0.9509 ± 0.0145	0.9506 ± 0.0124

of stability. The very low standard deviations, ranging from roughly 0.6% to 1.4%, indicate that the system's ability to learn the binary trigger task is negligibly dependent on the specific activities assigned to the partitions. This finding confirms that the proposed LAT architecture is robust and capable of learning to distinguish any arbitrary subset of activities from the background class, provided there is sufficient training data. The approach does not rely on specific kinematic features unique to hand-washing or eating but generalizes well to arbitrary binary classification tasks within the HAR domain.

5.4.2 Semantic Template Recognition

After characterizing the LAT system, which acts as an energy-efficient gatekeeper, we evaluate the effectiveness of the Siamese Network on the semantic template recognition methodology. The evaluation is twofold: first, we assess its performance in standard multi-class classification scenarios, and second, we investigate its capability in Generalized Zero-Shot Activity Recognition (GZSAR).

Multi-Class Classification Results

We first benchmarked the semantic template approach on the standard classification task. The datasets were randomly split into training (75%) and testing (25%) subsets. The training set was used to generate positive/negative pairs for training the Siamese network and subsequently to construct the semantic templates, while the test set was used to calculate performance metrics. Table 5.8 compares three variants of the Siamese model, varying in complexity from #0 to #2, against two state-of-the-art solutions designed for constrained devices: a Vanilla LSTM (Contoli et al. [352]) and a TinyHAR CNN (Nooruddin et al. [362]).

The Siamese-based approach demonstrates competitive classification accuracy while achieving superior energy efficiency. Our model outperforms the Vanilla LSTM in the RealWorld and WISDM datasets. While the TinyHAR CNN achieves the highest accuracy on WISDM (approx. 10% gap) and slightly outperforms our model on UCI-HAR (2% gap), this comes at a significant energy cost. While the TinyHAR CNN uses few parameters, its execution on the ESP32 platform is inefficient, consuming up to 190 mJ per inference due to a ≈ 1000 ms inference time. Similarly, the Vanilla LSTM consumes between 27 and 36 mJ. In stark contrast, Model #1 consumes only 11.92 mJ, a reduction of approximately 50% compared to the LSTM and over one order of magnitude compared to the CNN. Consequently, Model #1 was selected for all subsequent experiments, as it offers the optimal trade-off, halving the energy consumption of Model #0 with negligible performance loss.

Table 5.9 summarizes the classification performance obtained by varying the aggregation function used to build the semantic templates. For each dataset, the best-performing configuration is highlighted in bold. In the case of the Similarity Conditioned Average (SCA), the reported value corresponds to the best result achieved across different similarity thresholds (S_{th}). The optimal threshold is indicated as a subscript of the SCA label. For example, the best threshold for the RealWorld dataset was 0.8, while a value of 0.9 was optimal for WISDM. Overall, the results show that the Sliding Window Average (SWA) aggregation tends to yield the lowest classification accuracy on the WISDM and UCI-HAR datasets. In contrast, the SCA approach with a similarity threshold of 0.8 provides the most favorable results for the RealWorld dataset. Despite these differences, the performance gap among the three semantic template construction strategies never exceeds one percentage point, indicating that the overall system is only marginally sensitive to the choice of aggregation function.

Table 5.8: Classification performance comparison obtained on the multi-class problem for the Siamese Models and the state-of-the-art models.

model	dataset	accuracy	precision	recall	f1-score	# params	energy [mJ]
#0	RealWorld	**86.85** ± 0.12	**88.89** ± 0.11	**88.06** ± 0.08	**88.30** ± 0.11	18,070	19.73
	WISDM	71.37 ± 0.67	73.52 ± 0.75	74.12 ± 0.67	73.43 ± 0.67		
	UCI-HAR	77.48 ± 0.60	79.29 ± 0.89	**79.28** ± 0.61	79.03 ± 0.84		
#1	RealWorld	86.07 ± 0.86	87.35 ± 0.94	86.69 ± 0.69	86.86 ± 0.76	9,831	**11.92**
	WISDM	72.54 ± 0.86	71.75 ± 0.88	72.26 ± 0.87	71.58 ± 0.90		
	UCI-HAR	72.11 ± 0.99	74.62 ± 0.80	74.53 ± 0.89	74.42 ± 0.94		
#2	RealWorld	84.60 ± 0.60	86.06 ± 0.98	85.43 ± 0.56	85.54 ± 0.69	3,526	2.95
	WISDM	67.51 ± 0.85	66.53 ± 0.87	67.16 ± 0.80	66.35 ± 0.82		
	UCI-HAR	64.74 ± 0.84	67.90 ± 0.81	67.96 ± 0.76	67.71 ± 0.74		
Contoli et al. [352]	RealWorld	86.71	88.58	87.60	87.82	7,880	27.23
	WISDM	71.74	73.41	71.38	70.99	10,450	36.11
	UCI-HAR	**82.24**	**80.54**	79.12	**80.45**	8,651	29.89
Nooruddin et al. [362]	WISDM	**84,80**	**83.50**	**84.80**	**83.30**	1,321	160.00
	UCI-HAR	78.20	70.7	78.20	78.30	3,974	190.00

Table 5.9: Classification performance for the multi-class problem varying the aggregation function (AA, SWA, SCA).

metric	RealWorld			WISDM			UCI-HAR		
	AA	SWA	$SCA_{0.8}$	AA	SWA	$SCA_{0.9}$	AA	SWA	$SCA_{0.8}$
accuracy	**86.07**	85.48	85.88	72.54	72.88	**72.89**	72.11	72.96	**72.97**
precision	87.35	87.51	**87.95**	71.75	**72.49**	72.11	74.62	**75.37**	75.31
recall	86.69	86.76	**87.11**	72.26	**72.59**	71.92	74.53	**75.29**	75.27
f1-score	86.86	86.99	**87.35**	71.58	**72.33**	71.83	74.42	**75.28**	75.23

Generalized Zero-Shot Activity Recognition

To assess the system's ability in recognizing activities it has never encountered during training, we conducted a series of GZSAR experiments. The dataset partitioning strategy is critical for a valid GZSAR assessment. The entire dataset was divided into a training/template Generation Set (80%) and a hold-out test Set (20%). The activities were categorized into *seen* classes that were used to train the Siamese network and *unseen* classes that were never exposed to the network during training. In the case of the RealWorld dataset, the selected unseen classes are *laying, climbing up, walking*, for the WISDM *walking, drinking, writing*, and for the UCI-HAR *walking, walking downstairs, standing*. To prevent the model from overfitting to specific samples, the seen classes were further split using 80% to train the Siamese Network, while the remaining 20% were reserved strictly for building the semantic templates. For unseen classes, templates were built using samples from the 80% partition, but crucially, none of these samples were used to update the network weights.

Table 5.10 summarizes the classification accuracy. The table reports metrics for classifying seen classes only ($A_{\mathcal{S}\to\mathcal{S}}$), unseen classes only ($A_{\mathcal{U}\to\mathcal{U}}$), and the challenging GZSAR scenario where the model must distinguish between both ($A_{\mathcal{S}\to\mathcal{C}}$ and $A_{\mathcal{U}\to\mathcal{C}}$). The system

Table 5.10: Best classification accuracy measured in the GZSAR experiments for the three reference datasets.

metric	RealWorld			WISDM			UCI-HAR		
	AA	SWA	$SCA_{0.8}$	AA	SWA	$SCA_{0.8}$	AA	SWA	$SCA_{0.8}$
$A_{\mathcal{S}\to\mathcal{S}}$	**89.78**	89.74	89.30	**76.19**	75.49	74.80	91.70	**91.75**	91.47
$A_{\mathcal{U}\to\mathcal{U}}$	77.07	77.16	**77.49**	**91.20**	90.53	90.53	92.10	92.32	**92.84**
$A_{\mathcal{S}\to\mathcal{C}}$	**75.13**	69.09	69.39	67.22	**67.97**	67.68	66.83	66.76	**68.98**
$A_{\mathcal{U}\to\mathcal{C}}$	52.33	57.77	**60.73**	**69.76**	64.16	64.64	70.51	**70.97**	70.95
HM	61.69	62.92	**64.77**	**68.47**	66.01	66.13	68.62	68.88	**69.95**

exhibits remarkable capability in recognizing activities it has never seen. For instance, in the UCI-HAR dataset, the accuracy on unseen classes alone ($A_{\mathcal{U}\to\mathcal{U}}$) reaches 92.84%. In the full generalized zero-shot setting, identifying both seen and unseen concurrently, the model achieves up to 75% accuracy for seen classes on the RealWorld dataset and 71% for

unseen classes on the UCI-HAR dataset. These results compare very favorably with recent benchmarks. For example, Wang et al. [361] reported GZSAR accuracies ranging from 39–58% for seen classes and 37–52% for unseen classes. Our approach significantly outperforms these figures. Unlike standard classification, GZSAR performance is sensitive to the template aggregation method. The Similarity Conditioned Average (SCA) generally outperforms others, especially on RealWorld and UCI-HAR, when considering the harmonic mean of seen and unseen accuracies. However, for WISDM, the Arithmetic Average (AA) remains competitive.

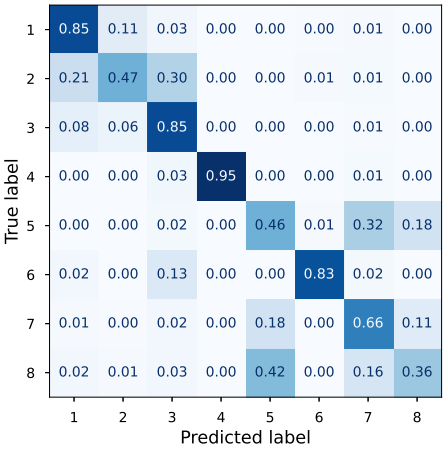
Figure 5.12 presents the confusion matrices obtained in a GZSAR experiment for the three datasets. The clear diagonal structure confirms that the system effectively discriminates between both seen and unseen classes. For example, in UCI-HAR (Figure 5.12 b), the unseen walking activity (index 1) is correctly classified 86% of the time. Moreover, we observed a phenomenon where certain unseen classes act as attractors for semantically similar seen classes. A notable example is in the RealWorld dataset (Figure 5.12 a), where the unseen class climbing up (index 5) attracts approximately 42% of the samples from the seen class running (index 8). This misclassification is understandable given the kinematic similarities between high-intensity locomotion activities.

In summary, the results indicate that the semantic template strategy provides an effective solution for class-incremental learning on resource-limited devices, enabling the system to incorporate new activities without the need for complete retraining.

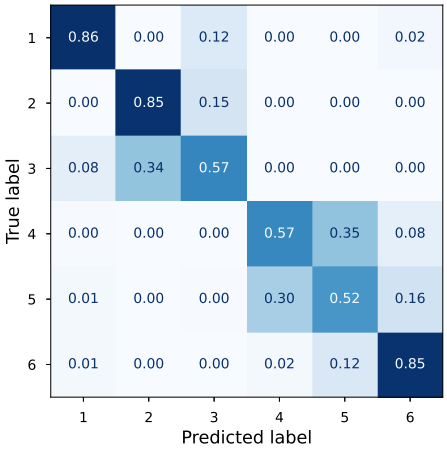
Sensitivity to the Configuration Parameters

To ensure the robustness of the GZSAR framework, we conducted a sensitivity analysis on two critical hyperparameters: the template calibration factor (α) and the similarity threshold (Sth). The calibration factor α plays a pivotal role in balancing the classifier's bias between seen and unseen classes. To evaluate its impact, we computed the seen-unseen accuracy curve for varying values of α, following the methodology proposed by Chao et al. [363]. Figure 5.13 illustrates these curves for the three reference datasets. The curves exhibit a classic trade-off behavior. Increasing α reduces the similarity score required for seen classes, thereby biasing the system towards unseen classes (increasing $A_{\mathcal{U}\to\mathcal{C}}$ at the expense of $A_{\mathcal{S}\to\mathcal{C}}$). Conversely, decreasing α favors the seen classes. To encapsulate this trade-off in a single measure, we computed the Area Under the Seen–Unseen Curve (AUSUC). Higher values indicate a better balance in maintaining accuracy for both seen and unseen classes. Among the datasets, UCI-HAR attains the highest AUSUC, around 0.71, highlighting the system's robust generalization performance on this benchmark.

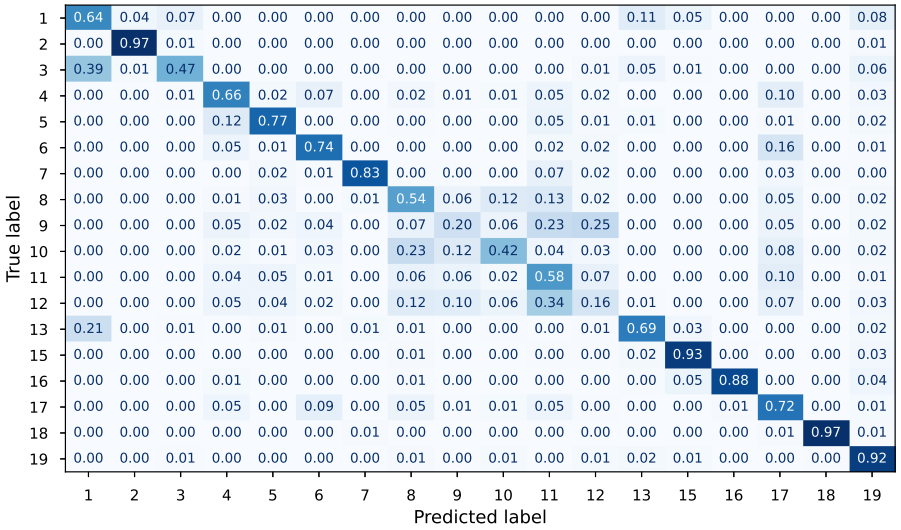
To pinpoint the optimal α value, we utilized a Pareto optimization approach. Figure 5.14 plots the misclassification rates of seen activities ($MCR_{\mathcal{S}\to\mathcal{C}}$) against unseen ones ($MCR_{\mathcal{U}\to\mathcal{C}}$). The optimal operating points, highlighted with circles, are those minimizing both error rates simultaneously. In particular, on both WISDM and UCI-HAR, the best alpha value is 0.002, while for RealWorld it is 0.001. The extremely low magnitude of these optimal values suggests that our approach is inherently robust against *seen-class bias*, requiring very little calibration to treat unseen classes fairly.



(a) *RealWorld*, 1: Standing, 2: Laying, 3: Sitting, 4: Jumping, 5: Climbing Up, 6: Climbing Down, 7: Walking, 8: Running. (Unseen classes: 2,5,7).



(b) *UCI-HAR*, 1: Walking, 2: Walking Upstairs, 3: Walking Downstairs, 4: Sitting, 5: Standing, 6: Laying. (Unseen classes: 1,3,5).



(c) *WISDM*, 1: Walking, 2: Jogging, 3: Climbing Stairs, 4: Sitting, 5: Standing, 6: Typing, 7: Brushing Teeth, 8: Eating Soup, 9: Eating Chips, 10: Eating Pasta, 11: Drinking, 12: Eating Sandwich, 13: Kicking, 15: Catch, 16: Dribbling, 17: Writing, 18: Clapping, 19: Folding. (Unseen classes: 1,2,6).

Figure 5.12: Confusion matrices of the three datasets obtained with $Sth = 0.95$ and $\alpha = 0.002$.

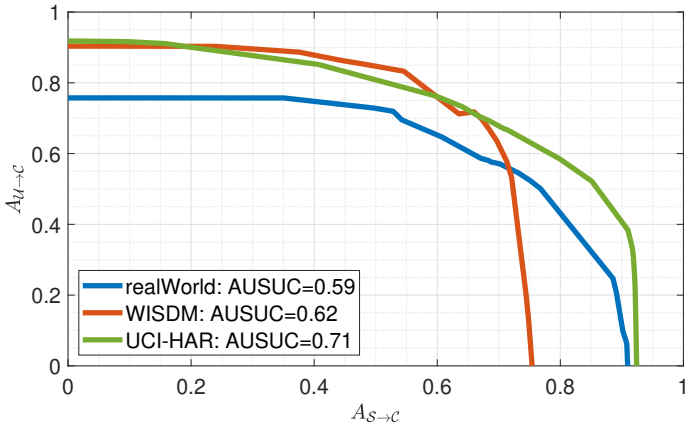


Figure 5.13: The seen-unseen accuracy curve calculated for the three reference datasets when varying α.

The second key parameter is the similarity threshold (Sth), which governs the Similarity Conditioned Average (SCA) aggregation function. This threshold determines which samples are considered representative enough to contribute to a class template. Figure 5.15 presents the Pareto chart for Sth, plotting the trade-off between seen and unseen misclassification rates. Across all three datasets, a value of $Sth = 0.8$ emerges as the universally optimal value. This threshold implies that templates should be constructed using a strictly homogeneous set of samples to ensure representativeness. However, pushing the threshold higher (i.e., > 0.8) is counterproductive, as it risks creating rigid templates that overfit the training instances and fail to generalize to natural variations in human movement.

5.4.3 Tiny-HAR Transformer

While the LAT and semantic template approaches focus on efficiency and few-shot learning flexibility, respectively, the current trend in Deep Learning is dominated by the Transformer architectures. Known for their ability to model long-range dependencies via self-attention mechanisms, Transformers represent the cutting edge of sequence modeling. However, deploying them on constrained devices like the ESP32 presents a unique set of challenges compared to the CNN or LSTM architectures previously discussed. In the following, we explore the characterization of the Tiny-HAR-Transformer architecture, considering the trade-offs between the theoretical representational power of Transformers and the strict hardware limitations of low-power wearable devices.

The largest viable configuration of the Tiny-HAR-Transformer that fit within the memory constraints of the ESP32 consisted of two encoder blocks, each featuring a multi-head attention layer with two heads, followed by fully connected layers with 512 and 256 neurons, respectively. Table 5.11 benchmarks this Transformer model against state-of-the-art full-fledged classifiers running on PC and the Tiny-Vanilla models presented by Contoli et al. [352]. As expected, the full-fledged models achieve more than 90% accuracy across all

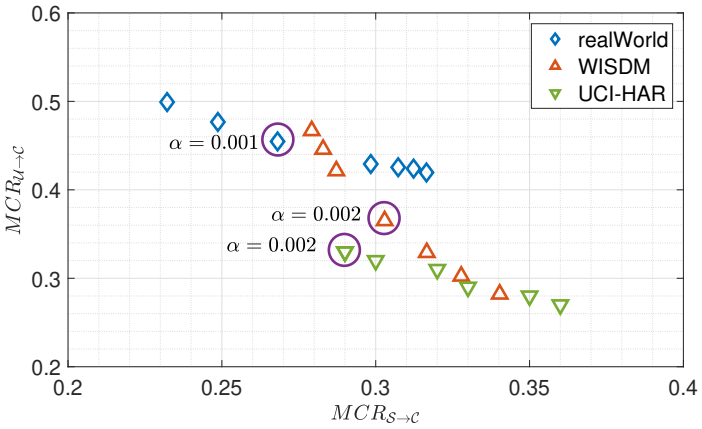


Figure 5.14: Pareto chart comparing the misclassification rate of seen activities ($MCR_{\mathcal{S}\to\mathcal{C}}$) with unseen activities ($MCR_{\mathcal{U}\to\mathcal{C}}$) for the three datasets when varying α.

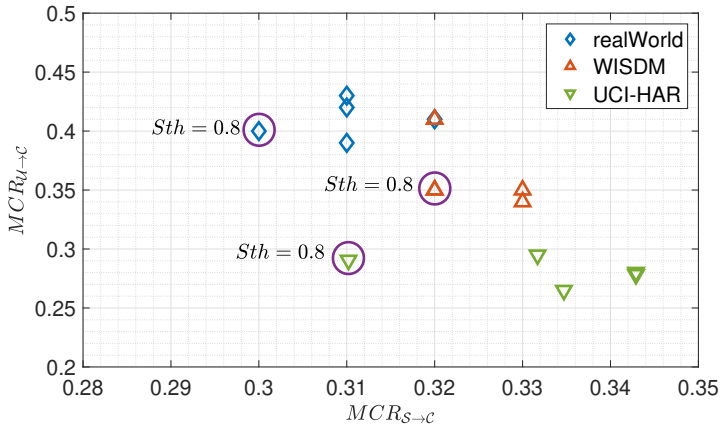


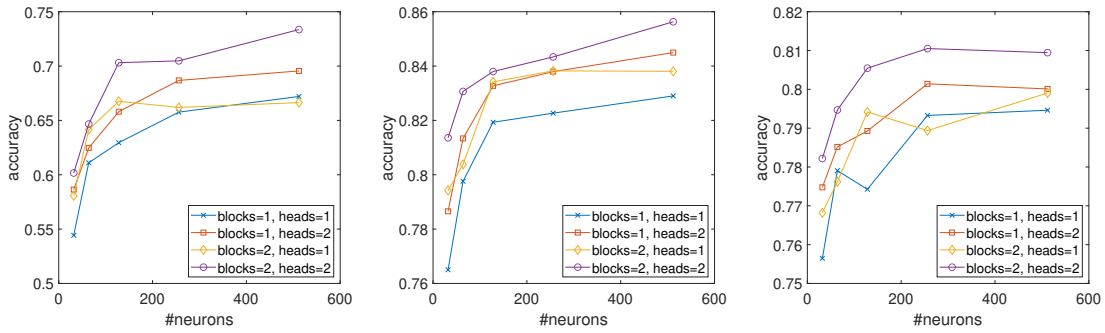
Figure 5.15: Pareto chart comparing the misclassification rate of seen activities ($MCR_{\mathcal{S}\to\mathcal{C}}$) with unseen activities ($MCR_{\mathcal{U}\to\mathcal{C}}$) for the three datasets when varying Sth

Table 5.11: Classification accuracy of the selected classifier on top of the reference datasets.

	CNN [353]	CNN_LSTM [349]	Trans. [364]	Tiny-Vanilla [352]	Tiny-HAR-Trans.
WISDM	0.936	0.921	0.938	0.873	0.734
RealWorld	0.956	0.937	0.946	0.921	0.856
Watch_HAR	0.923	0.906	0.916	0.880	0.811

datasets, with CNNs dominating on RealWorld and Watch_HAR, and Transformers leading on WISDM. In the constrained domain, the Tiny-Vanilla model currently outperforms the Tiny-HAR-Transformer by more than 7% on all datasets and up to $\approx 14\%$ on WISDM. The performance gap is not necessarily due to the inferiority of the Transformer architecture itself, but rather the inability to deploy a sufficiently deep model on the ESP32. A deeper model can not be executed due to the heavy request for RAM memory. Unlike CNNs, which can often process data in-place layer by layer, the residual connections fundamental to Transformers require preserving a copy of the input tensor to be added to the layer's output. Considering that in sensor-based HAR, a standard 2-second window at 50 Hz with 6 channels (triaxial accelerometer + gyroscope) creates an input tensor of approximately 20 KB. Thus, storing multiple copies of this tensor for residual connections rapidly saturates the limited RAM of the ESP32, preventing the deployment of deeper, more expressive models. In contrast, the Tiny-Vanilla CNNs exploit Flash memory for weights and require less dynamic RAM, allowing Contoli et al. to deploy deeper networks.

To understand the scaling behavior of the Transformer on microcontrollers, we explored the hyperparameter space varying *num_blocks*, *num_heads*, and *neurons*. Figure 5.16 illustrates the accuracy trends across the three datasets. The results consistently show



(a) WISDM (b) RealWorld (c) Watch_HAR

Figure 5.16: Classification performance for different transformer configurations on the three reference datasets.

that "bigger is better". Across all datasets, the largest runnable configuration (2 blocks, 2 heads) yielded the highest accuracy. Increasing the number of neurons generally led to monotonic improvements in accuracy for WISDM and RealWorld. The trends suggest that the model has not yet saturated its learning potential. If hardware constraints allowed for deeper networks with more blocks, the Transformer would likely close the performance gap

with the Tiny-Vanilla CNNs. The limitation is strictly hardware-bound, not architectural.

While accuracy is currently limited by RAM, the inference speed offers a promising counter-narrative. Figure 5.17 reports the on-device execution times for various Transformer configurations compared to the Tiny-Vanilla baseline. Despite the complexity of

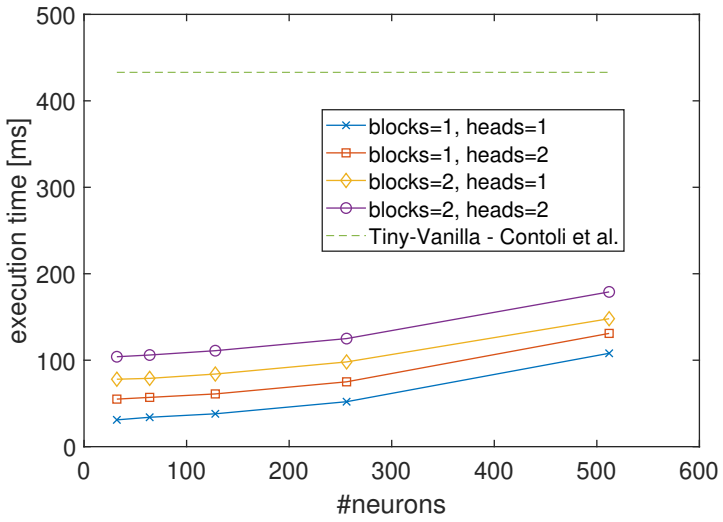


Figure 5.17: Execution times for different transformer configurations compared to the Tiny-Vanilla baseline.

the attention mechanism, the Tiny-HAR-Transformer configurations run approximately 3× faster than the Tiny-Vanilla models. This result highlights a crucial trade-off of transformers: they are faster but RAM-hungry, whereas the Tiny-Vanilla CNN/LSTMs are slower but RAM-efficient.

5.4.4 Data Imputation and Generation

Deploying full-scale Transformer models directly on low-power wearable devices remains a significant challenge due to memory constraints. However, the architectural strengths of Transformers, specifically their ability to model long-range dependencies and understand context, remain undeniable. While real-time edge execution is still maturing, these models can be powerfully applied in offline or server-side scenarios to address critical HAR challenges, such as handling data loss (imputation) and data scarcity (augmentation). In this scenario, we evaluate the performance of applying a BERT-based LLM architecture to inertial data. In particular, we show the result in two key tasks: (i) reconstructing missing signal chunks and (ii) synthetically generating realistic activity traces.

Imputation Performance

We first evaluated the model's ability to reconstruct missing chunks in accelerometer data by masking specific tokens and asking BERT to predict them. The model was tested both

with random and span token masking approaches, which were also used during the training phase, plus a deterministic distance positioning of the masked tokens. In the deterministic masking, once the position of the first token was randomly chosen, subsequent tokens were placed at fixed distances of one or two positions. In particular, BERT was used to predict different numbers of tokens (from 1 to 16) for each signal window. Figure 5.18 illustrates the mean prediction accuracy as the number of masked tokens increases. The highest accuracy,

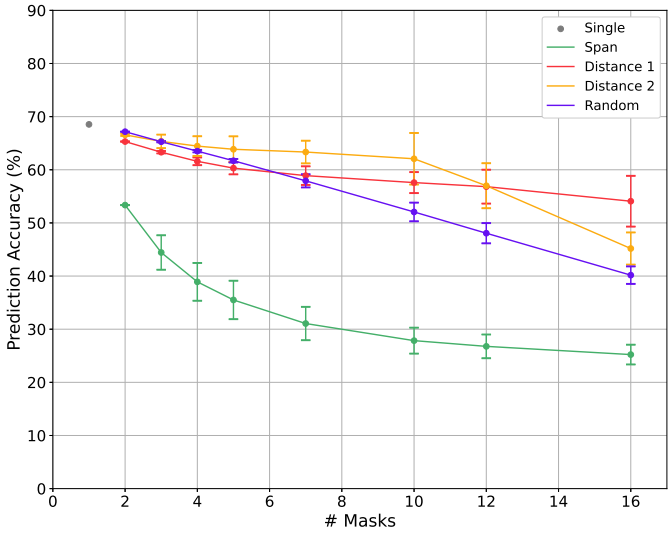


Figure 5.18: Average token prediction accuracy when increasing the number of masked tokens.

surpassing 70%, is achieved when predicting a single masked token, confirming that a broad, uncorrupted context maximizes the model's predictive power. In the case of random and deterministic masking, the performance degrades only marginally as more tokens are masked. This suggests the model is highly robust and capable of reconstructing the context even from sparse data. As expected, in the case of span masking, the reconstruction is significantly harder when a contiguous block of data is missing. Without immediate anchor points, accuracy stabilizes around 25%. However, the model still demonstrates generalization capabilities, attempting predictions even for 16 masked tokens, beyond the maximum of 12 seen during training.

While token accuracy measures exact matches, the MAE measures how close the reconstructed signal is to the original physical values (m/s^2). Figure 5.19 compares BERT against standard reference techniques. BERT outperforms all reference techniques in both experiments, especially for higher numbers of masked tokens. Particularly, the performance gap is most distinct in the difficult consecutive mask scenario. Here, standard interpolation fails, but BERT's generative nature allows it to hallucinate plausible signal trajectories, keeping the error low relative to the signal range from about -12 to 12 m/s^2.

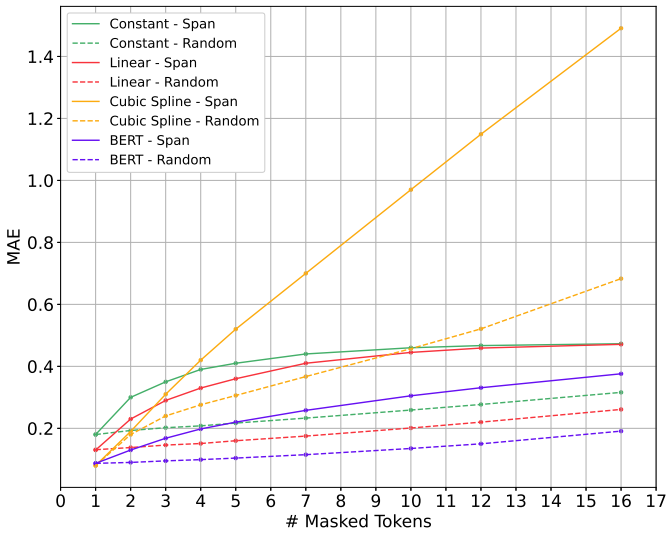


Figure 5.19: Reconstruction error of the accelerometer data, expressed as MAE, when varying the number of masked tokens, compared with the reference techniques.

Generation Performance

Next, we evaluated the model's ability to act as a generative engine to create entirely new synthetic training samples. We compared two generation methodologies: a multi-step generation based on iterative predictions and a single-step generation based on parallel predictions.

Table 5.12 compares the classification accuracy achieved by the state-of-the-art model on signals generated using both multi-step and single-step methodologies. For comparison, the reference accuracy obtained on real traces is also provided. The results clearly

Table 5.12: Intra-dataset classification accuracy for synthetic traces generated with multi-step and single-step methodologies.

	Accuracy		
Activity	**multi-step**	**single-step**	**Reference**
Walking	0.646	0.862	0.975
Jogging	0.909	0.912	0.940
Sitting	0.966	0.972	0.969
Standing	0.954	0.983	0.991
Stairs up	0.372	0.329	0.468
Stairs down	0.460	0.403	0.533
Average accuracy	**0.771**	**0.843**	**0.909**

demonstrate the superiority of the single-step generation approach, which attains an average accuracy of approximately 84%. Only 6 percentage points below the reference value for real data. A per-class analysis reveals that the static activities, like sitting or standing, are reproduced with high fidelity, yielding accuracy scores nearly identical to the reference. Conversely, more complex dynamic activities, specifically stairs up and stairs down, exhibit a performance gap of approximately 14 percentage points. However, it is important to note that these activities are inherently difficult to distinguish, as evidenced by their lower classification scores even in the reference scenario.

To assess how the fidelity of the synthetic traces evolves over time, we conducted a second series of experiments varying the length of the signal generated by the BERT model using the single-step methodology. We generated sequences ranging from 8 to 64 tokens, corresponding to synthetic accelerometer durations of 1.28s to 10.24s. Figure 5.20 illustrates the per-class accuracy as a function of the generated trace length. The observed

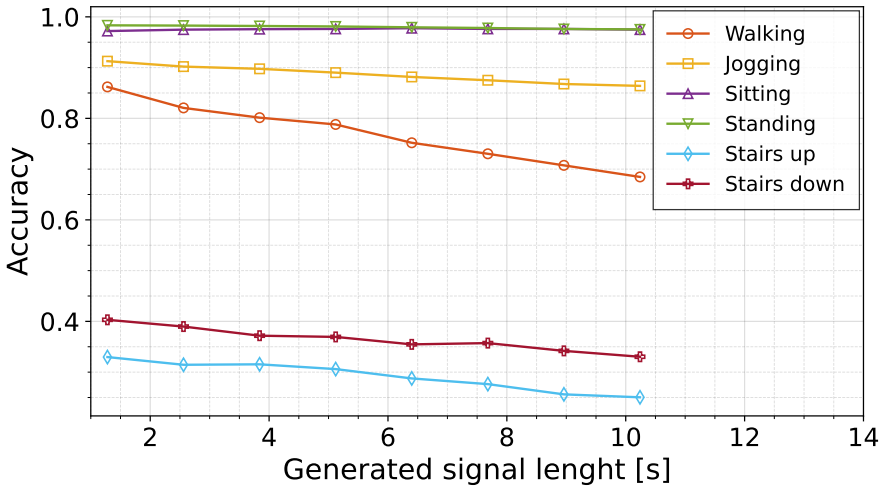
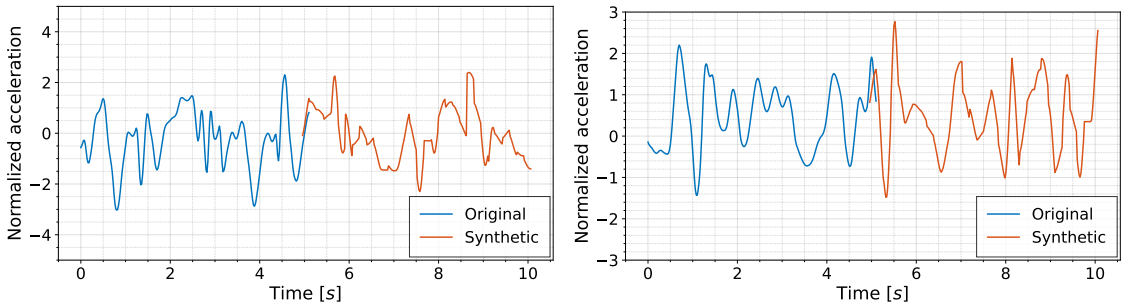


Figure 5.20: Per-class accuracy as a function of the generated trace length with the single-step methodology.

trends reveal activity-dependent behavior. For less dynamic activities, accuracy remains largely stable regardless of the length of the generated sequence. In contrast, for more dynamic activities, such as walking, accuracy decreases almost linearly as the synthetic window grows. A key point is that the BERT model's context window spans 5.12 seconds. Any generation extending beyond this period relies solely on previously generated data, as the original real inputs are no longer present. The model's ability to maintain coherent patterns in this fully synthetic regime demonstrates its strong generative capability.

Figure 5.21 offers a qualitative assessment of signal fidelity for the *jogging* and *stairs down* activities. The synthetic traces (red line) closely follow the real accelerometer signals (blue line), especially in the initial samples. Moreover, the model successfully reproduces the distinctive kinematic signatures of both activities, confirming that it captures the essential structure of human motion.



(a) Jogging (b) Stairs down

Figure 5.21: Qualitative comparison of generated signals (red line) versus real accelerometer data (blue line) used as input context.

The last evaluations concern the generalization capability of the approach by performing a cross-dataset validation. In this scenario, the model previously trained on one dataset was tasked with generating synthetic traces based on context provided by a completely different dataset with unknown subjects, devices, and collection protocols. Table 5.13 summarizes the classification accuracy of the synthetic traces as a function of the generation length, alongside the reference accuracy obtained on real data. The results demonstrate

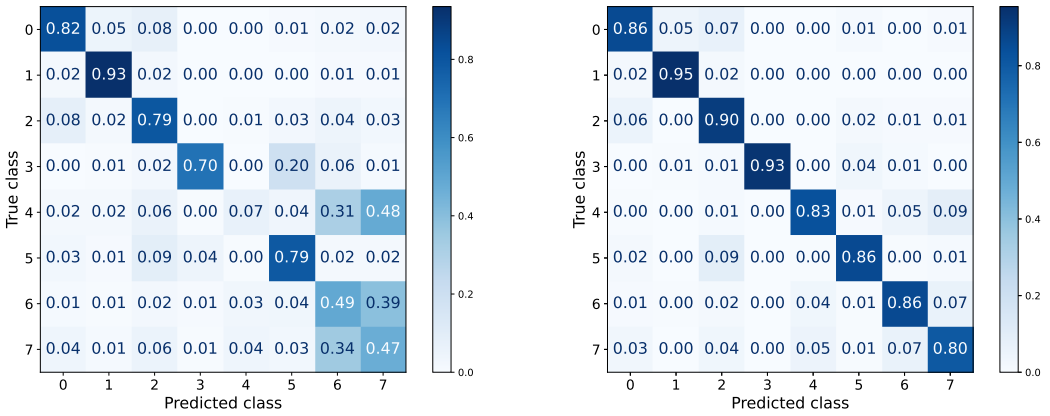
Table 5.13: Cross-dataset classification accuracy for synthetic traces when varying the lengths of the signal.

	Generated signal length [s]				
Class	**1.28**	**2.56**	**3.84**	**5.12**	**Reference**
Standing	0.833	0.825	0.820	0.818	0.855
Lying	0.934	0.934	0.934	0.934	0.954
Sitting	0.822	0.805	0.799	0.794	0.902
Jumping	0.747	0.735	0.716	0.695	0.932
Climbing up	0.053	0.063	0.064	0.071	0.830
Climbing down	0.812	0.799	0.793	0.788	0.859
Walking	0.558	0.527	0.502	0.491	0.857
Running	0.467	0.472	0.472	0.468	0.804
Average accuracy	**0.649**	**0.641**	**0.636**	**0.633**	**0.869**

the model's remarkable capacity to generate high-fidelity data, even when seeded with context from a previously unseen domain. Static and semi-static activities such as standing, sitting, and notably climbing down, are synthesized with high distinctiveness, achieving accuracy scores comparable to the reference system. Furthermore, activities that are typically challenging in cross-domain scenarios, such as laying and jumping, were synthesized with surprising realism, achieving top classification accuracies of approximately 93% and 75%, respectively. However, a significant anomaly is observed regarding the climbing up activity (highlighted in red), where accuracy drops to negligible levels ($\approx$ 5-7%). This is

counterintuitive, particularly given that the complementary activity, climbing down, was generated successfully.

To investigate the root cause of this misclassification, we analyzed the confusion matrices for both synthetic and real signals, presented in Figure 5.22. The matrices reveal



(a) Synthetic data (b) Real data

Figure 5.22: Confusion matrices summarizing the classification results of synthetic and original signals for the cross-validation dataset. Activities: standing (0), lying (1), sitting (2), jumping (3), climbing up (4), climbing down (5), walking (6), and running (7).

that while activities belonging to classes from 0 to 3 are correctly identified, the synthetic samples for class 4 (*climbing up*) are almost systematically misclassified as *walking* or *running*. This indicates that the BERT model, limited by the nuances captured in its training set, failed to distinguish the specific vertical kinematics of climbing from generic locomotion. Consequently, it produced a flattened or less sophisticated version of the activity that the classifier interpreted as simple walking or running. Similarly, a significant overlap is observed between *walking* and *running* in the synthetic data, with mutual confusion occurring nearly 50% of the time. This suggests that while the generative model captures the general periodicity of locomotion, it struggles to reproduce the specific intensity and distinctive ground-contact characteristics that differentiate these two activities in a cross-dataset setting.

5.5 Summary

The methodology defined in this chapter transforms the inertial data from wearable devices into a verifiable source of truth for hygiene management. This agent-centric approach establishes a dual operational value: it not only validates the remediation efforts performed by cleaning staff but also quantifies the contamination load generated by occupants. By distinguishing between sedentary behavior and high-intensity movement, the system provides a more accurate proxy for dust resuspension and biological dispersion than simple

presence counters, allowing facility managers to align cleaning schedules with the actual accumulation of risk.

The experimental results confirm that this level of insight can be achieved within the tight power constraints of wearable devices. The deployment of the Lightweight Accurate Trigger (LAT) proved critical for sustaining long-term autonomy. By using a hierarchical control scheme, where a low-power gatekeeper screens general motion before engaging the main classifier, the system achieved energy savings of up to 95% in real-world conditions compared to continuous monitoring baselines. Pareto optimization indicated that the 2Conv_1D architecture strikes the optimal balance, minimizing false negatives while keeping power consumption safely below the threshold required for continuous 24/7 operation.

Once an activity of interest is detected, the system overcomes the limitations of conventional classifiers through Semantic Template matching. The Siamese Neural Network achieved 92% accuracy in Generalized Zero-Shot Activity Recognition (GZSAR), successfully identifying cleaning tasks not seen during training. This approach significantly reduces the maintenance burden, as new hygiene protocols can be incorporated by simply updating lightweight reference templates rather than retraining the full model. Additionally, this metric-learning strategy proved highly efficient, cutting energy consumption by 50% compared to standard LSTM implementations.

Finally, the exploration of Transformer-based architectures revealed that treating human motion as a "language" offers significant advantages in signal resilience and inference speed. Despite the memory constraints of microcontrollers like the ESP32, the optimized Tiny-HAR-Transformer demonstrated an inference speed three times faster than recurrent architectures. Moreover, the BERT-based masking strategy outperformed traditional statistical interpolation in reconstructing corrupted data streams, effectively "hallucinating" coherent kinematic trajectories to fill signal gaps. However, the deployment of these advanced architectures on microcontrollers remains strictly constrained by a severe RAM bottleneck, which currently prevents the implementation of deeper and more expressive networks. Furthermore, in cross-dataset evaluations, the BERT model exhibited specific limitations in distinguishing fine vertical movements, such as climbing stairs. In these scenarios, subtle differences in kinematic intensity are often misclassified and confused with standard walking. These findings collectively establish that combining hierarchical resource management with semantic and generative modeling creates a robust, energy-efficient pillar for data-driven biological safety. Beyond hygiene management, this technology demonstrates strong transferability to other domains. The same zero-shot and hierarchical principles can be readily applied to monitor physical rehabilitation progress or to ensure industrial safety by verifying the correct usage of manual equipment in factory environments.

Chapter 6

Conclusions

Modern lifestyles have profoundly reshaped the relationship between humans and the indoor environment. Today, individuals spend up to 90% of their time inside enclosed spaces, including workplaces, educational facilities, healthcare buildings, and leisure environments. As a result, buildings can no longer be regarded merely as passive shelters. They have effectively become active determinants for public health, directly influencing exposure to biological, chemical, and physical risk factors. Despite this shift, the strategies traditionally adopted to manage indoor spaces have remained largely unchanged, still rooted in a paradigm that privileges visual order and perceived cleanliness over objective biological safety. Pathogenic microorganisms, including bacteria and viruses, are invisible to the naked eye and do not necessarily correlate with visible dirt. As a consequence, spaces that appear clean may still present significant biological risks to occupants.

High-touch "prime points", such as door handles, handrails, and elevator buttons, are well-known vectors for fomite-based transmission. However, surface contact represents only part of the problem. Respiratory emissions play an equally, if not more, important role. Every occupant continuously releases bioaerosols through breathing, speaking, coughing, and sneezing. These aerosolized particles, spanning a wide range of sizes, can remain suspended in the air for extended periods and gradually settle onto surrounding surfaces. Over time, this process leads to an invisible but persistent accumulation of microbial contamination both in the air and on surfaces. Fine droplets can penetrate deeply into the respiratory tract when inhaled, while larger particles deposit on floors, desks, and furnishings, effectively creating a hazardous biological "fallout". Dust particles can further act as carriers, enabling microorganisms to persist and spread through indirect contact. This dual pathway of airborne and surface contamination contributes to the development of Sick Building Syndrome (SBS) and underlies the alarming prevalence of Healthcare-Associated Infections (HAIs), which continue to cause tens of thousands of preventable deaths each year and impose a substantial burden on healthcare systems worldwide.

The traditional operational model adopted for cleaning and sanitization is poorly equipped to address these dynamics. Most facilities rely on rigid, time-based schedules that are disconnected from the actual usage and biological state of spaces. For instance, cleaning a restroom every fixed number of hours regardless of whether it has been heavily

used or is completely empty. Compounding this inefficiency is the lack of reliable verification and traceability. Cleaning activities are often documented through paper-based checklists signed by operators. Such systems are inherently vulnerable to errors, omissions, and intentional manipulation, and they offer no precise temporal or contextual information.

Recent research directions point toward a fundamental shift in this landscape. This vision advocates a transition from blind, schedule-driven cleaning toward adaptive, demand-based, and data-validated sanitization strategies. By embedding distributed sensing, edge intelligence, and predictive analytics directly into the building infrastructure, the facility itself becomes an active, cognitive system rather than a passive container of human activity. Within this framework, the building continuously perceives occupancy patterns, environmental conditions, and indicators of biological risk. It is capable of monitoring how human presence alters indoor air quality and surface contamination in real time. Triggering cleaning actions only when and where they are truly needed. This enables a decisive shift from reactive maintenance to proactive prevention, where safety is continuously assessed, interventions are justified by empirical data, and the well-being of occupants becomes a measurable and verifiable outcome.

In this broader context, the contributions of this thesis have been structured around three tightly interconnected methodological pillars. The first introduces the concept of an IoT virtual sensor. Rather than attempting to measure microbial contamination directly, the proposed approach exploits easily accessible environmental parameters, such as CO_2, volatile organic compounds, temperature, humidity, and particulate matter, as indirect indicators of biological load. These quantities are strongly influenced by human occupancy, ventilation efficiency, and microclimatic conditions, which are also the main drivers of microbial proliferation and dispersion. Through the adoption of Multi-Layer Perceptron (MLP) models, the system learns non-linear relationships between these environmental features and laboratory-measured microbial concentrations, enabling the estimation of airborne contamination levels. In this way, a standard IoT node is transformed into a virtual microbiological sentinel, capable of providing early warnings and actionable insights without the need for specialized and costly biological sensors.

The experimental validation demonstrates that this framework is not only conceptually sound but also technically viable on low-cost, commercially available hardware. The virtual sensor achieved performance well beyond initial expectations for systems based on commodity sensors. The MLP model successfully captured the complex dynamics linking environmental gases to bacterial load, reaching a coefficient of determination $R^2 = 0.92$. This result indicates that more than 90% of the variance in biological contamination can be explained using environmental proxies alone, effectively reducing the need for continuous manual sampling. The integration of data augmentation techniques based on Variational Autoencoders (VAE) further refined the estimation process, lowering the mean absolute error to approximately 50 CFU/m^3. This level of accuracy is more than sufficient to trigger preventive safety protocols well before critical risk thresholds are reached.

Ensuring that such a sensing infrastructure remains viable over long deployments required addressing energy sustainability as a first-class design constraint. To this end, the thesis developed and evaluated Predictive-Based Data Collection (PBDC) strategies tailored to resource-constrained IoT devices. The Deep Learning-Based Data Collection

(DLBDC) technique reduces radio traffic by suppressing transmissions whenever measured values remain consistent with locally generated predictions, effectively shifting the burden from communication to lightweight inference. The complementary Deep Learning Driven Sensing (DLDS) strategy goes a step further by allowing the device to autonomously predict the short-term evolution of environmental parameters and to extend its deep sleep intervals accordingly. By forecasting future samples, the node can remain inactive for multiple cycles while still guaranteeing accurate signal reconstruction at the cloud level. In the experiments, the DLBDC approach enabled the suppression of 82% of radio transmissions while maintaining a negligible reconstruction error of 1.4%. In parallel, the DLDS strategy dramatically extended device lifetime, achieving energy savings of up to 89% on ESP32 and Raspberry Pi Pico 2 W platforms. Together, these results demonstrate that continuous, intelligent, and biologically informed monitoring of indoor environments can be realistically deployed over existing wireless infrastructures, without imposing frequent maintenance or battery replacement, and while preserving both predictive accuracy and operational sustainability.

The second research pillar identifies human occupancy as the primary driver of hygienic degradation in indoor environments. To address this aspect, the thesis leverages computer vision. In the context of smart buildings, computer vision enables the observation of human presence and movement without relying on intrusive sensors, providing a powerful tool for understanding how spaces are actually used. Here, visual analysis is reinterpreted as a virtual occupancy sensor, capable of anonymously quantifying people counts, flow patterns, density levels, and dwell times. These metrics are essential for adaptive sanitization planning, as they directly reflect the intensity and spatial distribution of biological stress within a building. To ensure feasibility on edge platforms such as NVIDIA Jetson Nano or Google Coral, the proposed methodology adopts a lightweight tracking-by-detection pipeline. Object detection is combined with efficient tracking algorithms to maintain identities over time while minimizing CPU and GPU load. A key contribution is the Dynamic Inference Power Manager (DIPM). This intelligent control algorithm exploits the temporal redundancy inherent in video streams. In many real-world scenarios, consecutive frames differ only marginally. By continuously estimating scene dynamics, the DIPM autonomously decides when full inference can be safely skipped through adaptive frame skipping, without degrading tracking accuracy. This results in a closed-loop system that dynamically balances analytical precision against physical energy consumption.

A comparative analysis across different tracking families highlights a clear trade-off between accuracy and sustainability. While complex correlation-based trackers such as NvDCF achieve competitive tracking accuracy, they introduce an energy overhead of up to 288%, rendering them impractical for continuous deployment. In contrast, the IOU-based tracker maintains stable tracking performance with an energy increase limited to 9%, establishing it as a viable baseline for edge scenarios. The effectiveness of the DIPM was quantitatively validated through physical power measurements on embedded devices. In vehicular scenarios, characterized by structured and predictable motion patterns, the DIPM achieved an average energy saving of 36%. While in more pedestrian-dominated environments, the savings remained substantial at 21%. Crucially, these reductions in computational effort did not translate into a meaningful loss of analytical quality. Across all

evaluated scenarios, the error on primary tracking metrics (HOTA and MOTA) remained within 5%, well below thresholds that would affect downstream occupancy analytics or sanitization decisions. By combining lightweight tracking algorithms with the DIPM, the proposed approach enables continuous, real-time occupancy monitoring on edge devices while preserving both energy efficiency and operational accuracy, fully aligning with the vision of intelligent and resource-aware indoor environment management.

The third research pillar focuses on Sensor-Based Human Activity Recognition (HAR) methodologies and places human behavior at the center of the hygiene management problem. Within the scope of this thesis, HAR is not introduced merely as a tool for verifying healthcare workers' tasks, but as a fundamental mechanism for understanding how human activities actively shape the hygienic state of indoor environments. The type, intensity, and duration of activities performed in a space directly influence the generation and resuspension of bioaerosols, which subsequently settle on surfaces and contribute to environmental contamination. Consequently, recognizing "what people do" and "how they move" provides essential context for interpreting environmental measurements and for assessing how rapidly and in which areas hygienic conditions may degrade.

The thesis introduces a wearable-based HAR framework built on inertial measurement units (IMUs), which provide fine-grained motion data while remaining unobtrusive and privacy-preserving. A central challenge in this context is the limited energy budget of wearable devices, which constrains both continuous sensing and on-device inference. To address this issue, the proposed architecture adopts a hierarchical design called Lightweight Accurate Trigger (LAT). The LAT framework comprises two distinct, complementary models. The first is a lightweight trigger model that remains continuously active. Its role is not to perform detailed classification, but to detect generic motion patterns that are compatible with activities of interest. When the trigger detects a relevant activity signature, it activates the baseline classification model. This model is intentionally simple and computationally inexpensive, allowing it to operate persistently with minimal impact on battery life.

By decoupling activity detection from activity recognition, the LAT architecture significantly reduces unnecessary computation while preserving high recognition performance when detailed analysis is required. Experimental results demonstrate that this hierarchical design is a decisive factor for wearable efficiency. Acting as an intelligent filter, LAT reduces energy consumption by up to 95%, enabling continuous sensing while activating deep inference only at meaningful moments. These results position LAT as a practical enabler for long-term, real-world HAR deployments in professional environments.

Once the baseline model is activated, a second challenge emerges: ensuring robustness and scalability across a wide and evolving set of human activities. In real-world deployments, it is unrealistic to assume that all possible actions can be anticipated and labeled during training. To overcome this limitation, the thesis introduces a semantic abstraction layer based on Generalized Zero-Shot Activity Recognition (GZSAR). This approach relies on Siamese Neural Networks (SNN) to compare observed motion patterns against a set of semantic templates rather than fixed class labels. Activities are recognized through similarity relationships in an embedding space, making it possible to incorporate new or

previously unseen actions without retraining the entire model. This capability is particularly relevant in professional cleaning and facility management contexts, where procedures and tools may vary across sites and evolve over time. The effectiveness of this approach is confirmed experimentally. The SNN-based framework achieves a 92% accuracy in zero-shot recognition, demonstrating the ability to identify new or unseen sanitization protocols with performance comparable to that of fully supervised models. This level of generalization is particularly valuable in cleaning and facility management contexts, where procedures may vary across sites and evolve over time.

Finally, the research explores an advanced and forward-looking modeling paradigm by interpreting human motion as a structured sequence analogous to language. By applying Transformer-based architectures and Large Language Models (LLMs) like BERT to inertial data, motion signals are tokenized and processed as temporal "sentences" of movement. This representation enables powerful self-supervised learning strategies, supporting both the reconstruction of corrupted or incomplete sensor data and the generation of realistic synthetic motion traces. Results show that BERT-based approaches significantly outperform traditional statistical interpolation methods in signal reconstruction tasks. Importantly, the thesis also validates the feasibility of executing these transformer-based models on commonly used wearable platforms for sensor-based HAR, demonstrating that even large and expressive architectures can be adapted to resource-constrained devices through careful design and optimization. Moreover, optimized Tiny-HAR-Transformer models achieve inference speeds up to three times faster than recurrent architectures such as RNNs and LSTMs, while maintaining high predictive accuracy.

While the proposed framework offers a comprehensive solution for indoor hygiene management, certain limitations must be acknowledged to guide future research. Regarding the first pillar, the proxy-based approach bridges the invisibility of biological risk only indirectly. Although the achieved coefficient of determination is highly promising, the transferability of this virtual sensor to environments with atypical ventilation dynamics remains a constraint. Future developments will require more heterogeneous training datasets to ensure robust generalization across diverse environment configurations. Concerning the second and third pillars, the successful resolution of the trade-off between accuracy and energy autonomy inherently imposes a dependence on current hardware architectures. The future adoption of MCUs equipped with dedicated accelerators could render some of the proposed data suppression strategies less effective. This hardware evolution will likely shift the current equilibrium between onboard computation and data transmission, requiring new optimization paradigms. Finally, the proposed Cleaning 4.0 framework operates as a modular ecosystem. Its transferability to other domains, such as smart nursing (elderly care) or logistics, is highly feasible, provided that the core requirements of privacy-by-design and energy sustainability are strictly maintained. The findings of this thesis indicate that these foundational pillars are valid and applicable well beyond the original investigated scenarios.

In conclusion, this thesis demonstrates that Cleaning 4.0 is not a conceptual abstraction, but a technically viable and scalable paradigm for next-generation indoor hygiene management. Rather than presenting a collection of isolated experiments, this work proposes a unified framework where three research pillars operate in deep synergy, capturing

the full chain of biological risk generation, propagation, and mitigation within built environments. Specifically, the IoT virtual sensors (Pillar 1) identify *when* the potential biological risk increases in the air. The edge-based computer vision algorithms (Pillar 2) locate *where* occupancy has saturated specific areas, leading to the accumulation of contaminants on surfaces. Finally, the wearable HAR systems (Pillar 3) verify *how* cleaning activities have been executed and quantify the contamination load generated by occupant behavior.

The integration of these heterogeneous intelligences transforms cleaning from a fixed operational cost based on static schedules into a dynamic, verifiable service. By effectively balancing advanced AI capabilities with the strict energy constraints of embedded devices, this framework maximizes occupant safety, reduces the ecological impact of unnecessary chemical usage, and paves the way for safer, more efficient, and resilient smart buildings.

References

[1] C. L. Stergiou, A. P. Plageras, V. A. Memos, M. P. Koidou, and K. E. Psannis, "Secure monitoring system for iot healthcare data in the cloud," *Applied Sciences*, vol. 14, no. 1, p. 120, 2023.

[2] R. Pitarma, G. Marques, and B. R. Ferreira, "Monitoring indoor air quality for enhanced occupational health," *Journal of Medical Systems*, vol. 41, no. 2, 2016.

[3] G. Marques and R. Pitarma, "Monitoring health factors in indoor living environments using internet of things," in *World Conference on Information Systems and Technologies*, pp. 785–794, Springer, 2017.

[4] Å. THÖRN, "Case study of a sick building: Could an integrated biopsychosocial perspective prevent chronicity?," *The European Journal of Public Health*, vol. 10, no. 2, pp. 133–137, 2000.

[5] J. Madureira, I. Paciência, C. Pereira, J. P. Teixeira, and E. d. O. Fernandes, "Indoor air quality in portuguese schools: levels and sources of pollutants," *Indoor air*, vol. 26, no. 4, pp. 526–537, 2016.

[6] L. Stabile, M. Dell'Isola, A. Russi, A. Massimo, and G. Buonanno, "The effect of natural ventilation strategy on indoor air quality in schools," *Science of the Total Environment*, vol. 595, pp. 894–902, 2017.

[7] B. Stephens, P. Azimi, M. S. Thoemmes, M. Heidarinejad, J. G. Allen, and J. A. Gilbert, "Microbial exchange via fomites and implications for human health," *Current Pollution Reports*, vol. 5, no. 4, p. 198–213, 2019.

[8] M. Traverse and H. Aceto, "Environmental cleaning and disinfection," *Veterinary Clinics: small animal practice*, vol. 45, no. 2, pp. 299–330, 2015.

[9] C. E. Anderson and A. B. Boehm, "Transfer rate of enveloped and nonenveloped viruses between fingerpads and surfaces," *Applied and Environmental Microbiology*, vol. 87, no. 22, pp. e01215–21, 2021.

[10] Y. L. A. Kwok, J. Gralton, and M.-L. McLaws, "Face touching: A frequent habit that has implications for hand hygiene," *American Journal of Infection Control*, vol. 43, no. 2, p. 112–114, 2015.

[11] I. Bocicor, M. Dascalu, A. Gaczowska, S. Hostiuc, A. Moldoveanu, A. Molina, A.-J. Molnar, I. Negoi, and V. Racovita, "Wireless sensor network based system for the prevention of hospital acquired infections," *arXiv preprint arXiv:1705.03505*, 2017.

[12] Y. B. Jinila, J. J. Thomas, and B. P. Shan, "Internet of things enabled approach for hygiene monitoring in hospitals," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–5, IEEE, 2020.

[13] S. Tebrizcik, S. Ersöz, E. Duman, A. Aktepe, and A. K. Türker, "Optimization of cleaning and hygiene processes in healthcare using digital technologies and ensuring quality assurance with blockchain," *Applied Sciences*, vol. 15, no. 15, p. 8460, 2025.

[14] L. Yu, "Real-time monitoring of air pathogens in the icu with biosensors and robots," *Critical Care*, vol. 26, no. 1, 2022.

[15] H. Zhang and A. Lai, "Characteristic of real-time airborne bacteria measurement and the response of low-cost particulate matter sensor: a pilot study," *Journal of Building Engineering*, p. 113021, 2025.

[16] B. G. Mitchell, L. Hall, N. White, A. G. Barnett, K. Halton, D. L. Paterson, T. V. Riley, A. Gardner, K. Page, A. Farrington, *et al.*, "An environmental cleaning bundle and health-care-associated infections in hospitals (reach): a multicentre, randomised trial," *The lancet infectious diseases*, vol. 19, no. 4, pp. 410–418, 2019.

[17] C. A, "Sinners circle reloaded: A new concept to understand the real cleaning mechanism of dishwashing," in *FCFP 2025 Herausgeber: Fraunhofer Institut fur Verfahrenstechnik und Verpackung*, pp. 1–10, Fraunhofer Institut fur Verfahrenstechnik und Verpackung, 2025.

[18] A. Dramowski, M. Aucamp, A. Bekker, S. Pillay, K. Moloto, A. Whitelaw, M. Cotton, and S. Coffin, "Neoclean: a multimodal strategy to enhance environmental cleaning in a resource-limited neonatal unit," *Antimicrobial Resistance & Infection Control*, vol. 10, no. 1, p. 35, 2021.

[19] F. Azman, N. Saleh, F. Hashim, A. Sali, A. Ali, and A. Noor, "An iot-based hygiene monitoring system in the restroom," *IEEE Access*, 2025.

[20] C. Sánchez-Rodríguez, L. Capitán-Moyano, N. Malih, A. M. Yánez, M. Bennasar-Veny, O. Velasco-Roldán, O. Bulilete, and J. Llobera-Canaves, "Prevalence of musculoskeletal disorders among hotel housekeepers and cleaners: A systematic review with meta-analysis," *Musculoskeletal Science and Practice*, vol. 69, p. 102890, 2024.

[21] S. F. Bloomfield, M. Exner, C. Signorelli, K. Nath, and E. Scott, "The chain of infection transmission in the home and everyday life settings, and the role of hygiene in reducing the risk of infection," in *International scientific forum on home hygiene*, 2012.

[22] B. Santella, A. Donato, L. Fortino, V. Satriani, R. F. Ferrara, E. Santoro, W. Longanella, G. Franci, M. Capunzo, and G. Boccia, "Clean to prevent, monitor to protect: A scoping review on strategies for monitoring cleaning in hospitals to prevent hais," *Infectious Disease Reports*, vol. 17, no. 5, p. 120, 2025.

[23] R. Fontana, M. Buratto, A. Caproni, C. Nordi, M. Pappadà, B. Bandera, L. Vogli, C. Buffone, and P. Marconi, "Evaluating cleaning services in civil environments: Microbiological and life cycle analysis comparing conventional and sustainable methods," *Sustainability*, vol. 16, no. 2, p. 487, 2024.

[24] G. Marques and R. Pitarma, "An internet of things-based environmental quality management system to supervise the indoor laboratory conditions," *Applied Sciences*, vol. 9, no. 3, p. 438, 2019.

[25] A. Zivelonghi and A. Giuseppi, "Smart healthy schools: An iot-enabled concept for multi-room dynamic air quality control," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 24–31, 2024.

[26] R. Sreevas, R. Shanmughasundaram, and V. VRL Swami, "Development of an iot based air quality monitoring system," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10S, p. 23–28, 2019.

[27] R. Mumtaz, S. M. H. Zaidi, M. Z. Shakir, U. Shafi, M. M. Malik, A. Haque, S. Mumtaz, and S. A. R. Zaidi, "Internet of things (iot) based indoor air quality sensing and predictive analytic—a covid-19 perspective," *Electronics*, vol. 10, no. 2, p. 184, 2021.

[28] N. S. Baqer, H. A. Mohammed, A. Albahri, A. Zaidan, Z. Al-Qaysi, and O. Albahri, "Development of the internet of things sensory technology for ensuring proper indoor air quality in hospital facilities: Taxonomy analysis, challenges, motivations, open issues and recommended solution," *Measurement*, vol. 192, p. 110920, 2022.

[29] L. Yang, T. Yao, G. Liu, L. Sun, N. Yang, H. Zhang, S. Zhang, Y. Yang, Y. Pang, X. Liu, and X. Hou, "Monitoring and control of medical air disinfection parameters of nosocomial infection system based on internet of things," *Journal of Medical Systems*, vol. 43, no. 5, 2019.

[30] L. Gamberini, P. Pluchino, D. Bacchin, A. Zanella, V. Orso, S. Anna, and D. Mapelli, "Iot as non-pharmaceutical interventions for the safety of living environments in covid-19 pandemic age," *Frontiers in Computer Science*, vol. 3, 2021.

[31] F. Cunico, L. Capogrosso, A. Castellini, F. Setti, P. Pluchino, F. Zordan, V. Santus, A. Spagnolli, S. Cordibella, G. Gennari, *et al.*, "The post-pandemic effects on iot for safety: the safe place project," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–4, IEEE, 2023.

[32] G. Marques and R. Pitarma, "Iaq evaluation using an iot co2 monitoring system for enhanced living environments," in *World Conference on Information Systems and Technologies*, pp. 1169–1177, Springer, 2018.

[33] E. Rary, S. M. Anderson, B. D. Philbrick, T. Suresh, and J. Burton, "Smart sanitation—biosensors as a public health tool in sanitation infrastructure," *International Journal of Environmental Research and Public Health*, vol. 17, no. 14, p. 5146, 2020.

[34] M. E. Rana, K. Shanmugam, O. C. Xian, and R. Abdulla, "Smart hygiene solutions for university campuses: Harnessing internet of things (iot) for health and safety," in *AIP Conference Proceedings*, vol. 3161, p. 020040, AIP Publishing LLC, 2024.

[35] S. Chandra, S. Srivastava, and A. Roy, "Public toilet hygiene monitoring and reporting system," in *2018 IEEE SENSORS*, pp. 1–4, IEEE, 2018.

[36] P. Deshmukh, A. Mohite, H. Bhoir, R. Patil, and A. Bhonde, "Intelligent public toilet monitoring system using iot," in *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, pp. 1–6, IEEE, 2020.

[37] J. Zhou, C. M. Welling, M. M. Vasquez, S. Grego, and K. Chakrabarty, "Sensor-array optimization based on time-series data analytics for sanitation-related malodor detection," *IEEE transactions on biomedical circuits and systems*, vol. 14, no. 4, pp. 705–714, 2020.

[38] J. Patel, A. Shah, C. Rushali, J. Patel, and V. Ukani, "Smart lavatory solution: Integrating iot and deep learning models for enhanced hygiene," *Scalable Computing: Practice and Experience*, vol. 26, no. 3, pp. 1057–1070, 2025.

[39] N. Turman-Bryant, T. Sharpe, C. Nagel, L. Stover, and E. A. Thomas, "Toilet alarms: A novel application of latrine sensors and machine learning for optimizing sanitation services in informal settlements," *Development Engineering*, vol. 5, p. 100052, 2020.

[40] E. Lattanzi, L. Calisti, and V. Freschi, "Unstructured handwashing recognition using smartwatch to reduce contact transmission of pathogens," *IEEE Access*, vol. 10, pp. 83111–83124, 2022.

[41] Y. JR, *Making IT Work : A History of the Computer Services Industry*. The MIT Press, 2017.

[42] Y. Jiang, T. H. Tran, M. Collins, and L. Williams, "Development of internet of things and artificial intelligence for intelligent sanitation systems: A literature review," *Journal of Infrastructure, Policy and Development*, vol. 8, pp. 2572–7923, 2024.

[43] A. Zohourian, S. Dadkhah, E. C. P. Neto, H. Mahdikhani, P. K. Danso, H. Molyneaux, and A. A. Ghorbani, "Iot zigbee device security: A comprehensive review," *Internet of Things*, vol. 22, p. 100791, 2023.

[44] M. Alipio and M. Bures, "Current testing and performance evaluation methodologies of lora and lorawan in iot applications: Classification, issues, and future directives," *Internet of things*, vol. 25, p. 101053, 2024.

[45] P. Zhou, G. Huang, L. Zhang, and K.-F. Tsang, "Wireless sensor network based monitoring system for a large-scale indoor space: data process and supply air allocation optimization," *Energy and Buildings*, vol. 103, pp. 365–374, 2015.

[46] F. Sánchez-Rosario, D. Sánchez-Rodríguez, J. B. Alonso-Hernández, C. M. Travieso-González, I. Alonso-González, C. Ley-Bosch, C. Ramírez-Casañas, and M. A. Quintana-Suárez, "A low consumption real time environmental monitoring system for smart cities based on zigbee wireless sensor network," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 702–707, 2015.

[47] Y. S. Cho, H. R. Kim, H. S. Ko, S. B. Jeong, B. Chan Kim, and J. H. Jung, "Continuous surveillance of bioaerosols on-site using an automated bioaerosol-monitoring system," *ACS sensors*, vol. 5, no. 2, pp. 395–403, 2020.

[48] E. Kabir, A. Azzouz, N. Raza, S. K. Bhardwaj, K.-H. Kim, M. Tabatabaei, and D. Kukkar, "Recent advances in monitoring, sampling, and sensing techniques for bioaerosols in the atmosphere," *ACS Sensors*, vol. 5, no. 5, p. 1254–1267, 2020.

[49] Y. Bouabdallaoui, Z. Lafhaj, P. Yim, L. Ducoulombier, and B. Bennadji, "Predictive maintenance in building facilities: A machine learning-based approach," *Sensors*, vol. 21, no. 4, 2021.

[50] A. P. Pomè and M. Signorini, "Real time facility management: assessing the effectiveness of digital twin in the operation and maintenance phase of building life cycle," in *IOP Conference Series: Earth and Environmental Science*, vol. 1176, p. 012003, IOP Publishing, 2023.

[51] V. Villa, B. Naticchia, G. Bruno, K. Aliev, P. Piantanida, and D. Antonelli, "Iot open-source architecture for the maintenance of building facilities," *Applied Sciences*, vol. 11, no. 12, p. 5374, 2021.

[52] I. H. Sarker, "Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective," *SN Computer Science*, vol. 2, no. 5, p. 377, 2021.

[53] A. Ahmad and M. Alshurideh, "Digitization and iot-driven transformation of smart buildings," *International Journal of Management and Marketing Intelligence*, vol. 2, no. 1, p. 26–38, 2025.

[54] L. Rueda, K. Agbossou, A. Cardenas, N. Henao, and S. Kelouwani, "A comprehensive review of approaches to building occupancy detection," *Building and Environment*, vol. 180, p. 106966, 2020.

[55] M. Antonino, M. Nicola, D. M. Claudio, B. Luciano, and R. C. Fulvio, "Office building occupancy monitoring through image recognition sensors," *International Journal of Safety and Security Engineering*, vol. 9, no. 4, pp. 371–380, 2019.

[56] K. Sun, Q. Zhao, and J. Zou, "A review of building occupancy measurement systems," *Energy and Buildings*, vol. 216, p. 109965, 2020.

[57] B. W. Hobson, D. Lowcay, H. B. Gunay, A. Ashouri, and G. R. Newsham, "Opportunistic occupancy-count estimation using sensor fusion: A case study," *Building and environment*, vol. 159, p. 106154, 2019.

[58] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and buildings*, vol. 112, pp. 28–39, 2016.

[59] R. Priyadarshini and R. Mehra, "Quantitative review of occupancy detection technologies," *Int. J. Radio Freq*, vol. 1, no. 1, pp. 1–19, 2015.

[60] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "Iot-based occupancy monitoring techniques for energy-efficient smart buildings," in *2015 IEEE Wireless communications and networking conference workshops (WCNCW)*, pp. 58–63, IEEE, 2015.

[61] N. Moretti, J. D. Blanco Cadena, A. Mannino, T. Poli, and F. Re Cecconi, "Maintenance service optimization in smart buildings through ultrasonic sensors network," *Intelligent Buildings International*, vol. 13, no. 1, p. 4–16, 2020.

[62] Q. Huang, K. Rodriguez, N. Whetstone, and S. Habel, "Rapid internet of things (iot) prototype for accurate people counting towards energy efficient buildings.," *Journal of Information Technology in Construction*, vol. 24, 2019.

[63] A. N. Sayed, Y. Himeur, and F. Bensaali, "Deep and transfer learning for building occupancy detection: A review and comparative analysis," *Engineering applications of artificial intelligence*, vol. 115, p. 105254, 2022.

[64] E. Lattanzi, M. Donati, and V. Freschi, "Exploring artificial neural networks efficiency in tiny wearable devices for human activity recognition," *Sensors*, vol. 22, no. 7, p. 2637, 2022.

[65] Y. Hui, J. Lien, and X. Lu, "Early experience in benchmarking edge ai processors with object detection workloads," in *International Symposium on Benchmarking, Measuring and Optimization*, pp. 32–48, Springer, 2019.

[66] P. Puchtler and R. Peinl, "Evaluation of deep learning accelerators for object detection at the edge," in *KI 2020: Advances in Artificial Intelligence: 43rd German Conference on AI, Bamberg, Germany, September 21–25, 2020, Proceedings 43*, pp. 320–326, Springer, 2020.

[67] P. Kang and A. Somtham, "An evaluation of modern accelerator-based edge devices for object detection applications," *Mathematics*, vol. 10, no. 22, p. 4299, 2022.

[68] Y. Guo and L. Zhou, "Mea-net: a lightweight sar ship detection model for imbalanced datasets," *Remote Sensing*, vol. 14, no. 18, p. 4438, 2022.

[69] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1438–1452, 2011.

[70] Z. Zhao, Z. Jiang, N. Ling, X. Shuai, and G. Xing, "Ecrt: An edge computing system for real-time image-based object tracking," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pp. 394–395, 2018.

[71] H. Zhang, Z. Zhang, L. Zhang, Y. Yang, Q. Kang, and D. Sun, "Object tracking for a smart city using iot and edge computing," *Sensors*, vol. 19, no. 9, p. 1987, 2019.

[72] B. Blanco-Filgueira, D. Garcia-Lesta, M. Fernández-Sanjurjo, V. M. Brea, and P. López, "Deep learning-based multiple object visual tracking on embedded system for iot and mobile edge computing applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5423–5431, 2019.

[73] Y. Inoue, T. Ono, and K. Inouer, "Situation-based dynamic frame-rate control for online object tracking," in *2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC)*, pp. 119–122, IEEE, 2018.

[74] Y. Inoue, T. Ono, and K. Inoue, "Real-time frame-rate control for energy-efficient on-line object tracking," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 101, no. 12, pp. 2297–2307, 2018.

[75] F. Paissan, A. Ancilotto, and E. Farella, "Phinets: a scalable backbone for low-power ai at the edge," *ACM Transactions on Embedded Computing Systems*, vol. 21, no. 5, pp. 1–18, 2022.

[76] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, "Deepedgebench: Benchmarking deep neural networks on edge devices," in *2021 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 20–30, IEEE, 2021.

[77] A. Jevremovic, Z. Kostic, and D. Perakovic, "Energy-efficient edge intelligence: A comparative analysis of aiot technologies," *Mobile Networks and Applications*, pp. 1–9, 2023.

[78] H. Feng, G. Mu, S. Zhong, P. Zhang, and T. Yuan, "Benchmark analysis of yolo performance on edge intelligence devices," *Cryptography*, vol. 6, no. 2, p. 16, 2022.

[79] M. Danish, R. Verma, J. Brazauskas, I. Lewis, and R. Mortier, "Deepdish on a diet: low-latency, energy-efficient object-detection and tracking at the edge," in *Proceedings of the 5th International Workshop on Edge Systems, Analytics and Networking*, pp. 43–48, ASCE, 2022.

[80] J. A. Bhatti, M. Asif, S. Hussain, S. Wasi, and T. Rajab, "Smart street light for energy saving based on vehicular traffic volume," in *2022 Global Conference on Wireless and Optical Technologies (GCWOT)*, pp. 1–4, IEEE, 2022.

[81] H. Gomes, N. Redinha, N. Lavado, and M. Mendes, "Counting people and bicycles in real time using yolo on jetson nano," *Energies*, vol. 15, no. 23, p. 8816, 2022.

[82] M. Buric, M. Ivasic-Kos, and M. Pobar, "Player tracking in sports videos," in *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 334–340, IEEE, 2019.

[83] A. Savchenko, "Facial expression recognition with adaptive frame rate based on multiple testing correction," *Proceedings of Machine Learning Research*, 2023.

[84] J. Lee and K.-i. Hwang, "Yolo with adaptive frame control for real-time object detection applications," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36375–36396, 2022.

[85] Y. O. Sharrab and N. J. Sarhan, "Accuracy and power consumption tradeoffs in video rate adaptation for computer vision applications," in *2012 IEEE International Conference on Multimedia and Expo*, pp. 410–415, IEEE, 2012.

[86] K. Park and M. Kim, "Evso: Environment-aware video streaming optimization of power consumption," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 973–981, IEEE, 2019.

[87] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

[88] S. Lu, H. Lu, J. Dong, and S. Wu, "Object detection for uav aerial scenarios based on vectorized iou," *Sensors*, vol. 23, no. 6, p. 3061, 2023.

[89] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[90] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[91] D. Stadler, L. W. Sommer, and J. Beyerer, "Pas tracker: Position-, appearance-and size-aware multi-object tracking in drone videos," in *Computer Vision–ECCV 2020 Workshops*, pp. 604–620, Springer International Publishing, 2020.

[92] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[93] M. M. Dasari and R. K. S. S. Gorthi, "Iou–siamtrack: Iou guided siamese network for visual object tracking," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2061–2065, IEEE, 2020.

[94] B. Wei, H. Chen, S. Cao, Q. Ding, and H. Luo, "An iou-aware siamese network for real-time visual tracking," *Neurocomputing*, vol. 527, pp. 13–26, 2023.

[95] Y. Li, C. P. Chen, and T. Zhang, "A survey on siamese network: Methodologies, applications, and opportunities," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 994–1014, 2022.

[96] S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, M. Felsberg, and J. Matas, "Visual object tracking with discriminative filters and siamese networks: a survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6552–6574, 2022.

[97] A. Nandy, S. Haldar, S. Banerjee, and S. Mitra, "A survey on applications of siamese neural networks in computer vision," in *2020 International Conference for Emerging Technology (INCET)*, pp. 1–5, IEEE, 2020.

[98] I. A. Lungu, A. Aimar, Y. Hu, T. Delbruck, and S.-C. Liu, "Siamese networks for few-shot learning on edge embedded devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 488–497, 2020.

[99] A. A. Kareem, D. A. Hammood, A. A. Alchalaby, and R. A. Khamees, "A performance of low-cost nvidia jetson nano embedded system in the real-time siamese single object tracking: A comparison study," in *International Conference on Computing Science, Communication and Security*, pp. 296–310, Springer, 2022.

[100] A. A. Kareem, D. A. Hammood, and R. A. Khamees, "Optimizing siamese neural network with tensorrt on nvidia jetson nano," in *AIP Conference Proceedings*, vol. 2804, AIP Publishing, 2023.

[101] V. Chandrakanth, V. Murthy, and S. S. Channappayya, "Siamese cross-domain tracker design for seamless tracking of targets in rgb and thermal videos," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 1, pp. 161–172, 2022.

[102] A. Anwar, S. Nadeem, and A. Tanvir, "Edge-ai based face recognition system: Benchmarks and analysis," in *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, (Islamabad, Pakistan), pp. 302–307, IEEE, IEEE, 2022.

[103] N. Xu, C. Liu, Y. Feng, F. Li, X. Meng, Q. Lv, and C. Lan, "Influence of the internet of things management system on hand hygiene compliance in an emergency intensive care unit," *Journal of Hospital Infection*, vol. 109, pp. 101–106, 2021.

[104] A. Benudis, S. Stone, A. S. Sait, I. Mahoney, L. L. Price, A. Moreno-Koehler, E. Anketell, and S. Doron, "Pitfalls and unexpected benefits of an electronic hand hygiene monitoring system," *American journal of infection control*, vol. 47, no. 9, pp. 1102–1106, 2019.

[105] J. L. Santarpia, D. N. Rivera, V. L. Herrera, M. J. Morwitzer, H. M. Creager, G. W. Santarpia, K. K. Crown, D. M. Brett-Major, E. R. Schnaubelt, M. J. Broadhurst, *et al.*, "Aerosol and surface contamination of sars-cov-2 observed in quarantine and isolation care," *Scientific reports*, vol. 10, no. 1, pp. 1–8, 2020.

[106] K. J. Mckay, R. Z. Shaban, and P. Ferguson, "Hand hygiene compliance monitoring: do video-based technologies offer opportunities for the future?," *Infection, Disease & Health*, vol. 25, no. 2, pp. 92–100, 2020.

[107] E. G. McDonald, E. Smyth, L. Smyth, and T. C. Lee, "Hand hygiene "hall monitors": leveraging the hawthorne effect," *American Journal of Infection Control*, vol. 46, no. 6, pp. 706–707, 2018.

[108] C. Zhong, A. R. Reibman, H. A. Mina, and A. J. Deering, "Multi-view hand-hygiene recognition for food safety," *Journal of Imaging*, vol. 6, p. 120, 2020.

[109] V. Galluzzi, T. Herman, and P. Polgreen, "Hand hygiene duration and technique recognition using wrist-worn sensors," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks - IPSN '15*, (New York, New York, USA), pp. 106–117, ACM Press, 2015.

[110] M. Bal and R. Abrishambaf, "A system for monitoring hand hygiene compliance based-on internet-of-things," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1348–1353, IEEE, 2017.

[111] H. Li, S. Chawla, R. Li, S. Jain, G. D. Abowd, T. Starner, C. Zhang, and T. Plotz, "WristWash: Towards automatic handwashing assessment using a wrist-worn device," *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 132–139, 2018.

[112] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[113] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[114] S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[115] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2259–2322, 2021.

[116] E. Ramanujam, T. Perumal, and S. Padmavathi, "Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13029–13040, 2021.

[117] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based human activity recognition using hybrid lstm network," in *2020 IEEE SENSORS*, pp. 1–4, IEEE, 2020.

[118] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of internet of things (iot): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, 2020.

[119] S. Uma and R. Eswari, "Accident prevention and safety assistance using iot and machine learning," *Journal of Reliable Intelligent Environments*, pp. 1–25, 2021.

[120] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for internet of things," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1399–1417, 2018.

[121] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4921–4934, 2019.

[122] M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, "Deep learning for heterogeneous human activity recognition in complex iot applications," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[123] M. Alessandrini, G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Recurrent neural network for human activity recognition in embedded systems using ppg and accelerometer data," *Electronics*, vol. 10, no. 14, p. 1715, 2021.

[124] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "Jointdnn: an efficient training and inference engine for intelligent mobile cloud computing services," *IEEE Transactions on Mobile Computing*, 2019.

[125] A. Elsts, R. McConville, X. Fafoutis, N. Twomey, R. J. Piechocki, R. Santos-Rodriguez, and I. Craddock, "On-board feature extraction from acceleration data for activity recognition.," in *EWSN*, pp. 163–168, 2018.

[126] A. Khan, N. Hammerla, S. Mellor, and T. Plötz, "Optimising sampling rates for accelerometer-based human activity recognition," *Pattern Recognition Letters*, vol. 73, pp. 33–40, 2016.

[127] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, "Squeezing deep learning into mobile and embedded devices," *IEEE Pervasive Computing*, vol. 16, no. 3, pp. 82–88, 2017.

[128] K. Z. Haigh, A. M. Mackay, M. R. Cook, and L. G. Lin, "Machine learning for embedded systems: A case study," *BBN Technologies: Cambridge, MA, USA*, 2015.

[129] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (iot)," *Procedia Computer Science*, vol. 98, pp. 437–442, 2016.

[130] C. Gupta, A. S. Suggala, A. Goyal, H. V. Simhadri, B. Paranjape, A. Kumar, S. Goyal, R. Udupa, M. Varma, and P. Jain, "Protonn: Compressed and accurate knn for resource-scarce devices," in *International Conference on Machine Learning*, pp. 1331–1340, PMLR, 2017.

[131] P.-E. Novac, A. Castagnetti, A. Russo, B. Miramond, A. Pegatoquet, and F. Verdier, "Toward unsupervised human activity recognition on microcontroller units," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, pp. 542–550, IEEE, 2020.

[132] G. Bhat, Y. Tuncel, S. An, H. G. Lee, and U. Y. Ogras, "An ultra-low energy human activity recognition accelerator for wearable health applications," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–22, 2019.

[133] X. Wang, M. Magno, L. Cavigelli, and L. Benini, "Fann-on-mcu: An open-source toolkit for energy-efficient neural network inference at the edge of the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4403–4417, 2020.

[134] S. Disabato and M. Roveri, "Incremental on-device tiny machine learning," in *Proceedings of the 2nd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pp. 7–13, 2020.

[135] Z. Wang, Y. Wu, Z. Jia, Y. Shi, and J. Hu, "Lightweight run-time working memory compression for deployment of deep neural networks on resource-constrained mcus," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pp. 607–614, 2021.

[136] K. Wang, J. He, and L. Zhang, "Sequential weakly labeled multiactivity localization and recognition on wearable sensors using recurrent attention networks," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 355–364, 2021.

[137] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7598–7604, 2019.

[138] Y. L. Coelho, F. d. A. S. d. Santos, A. Frizera-Neto, and T. F. Bastos-Filho, "A lightweight framework for human activity recognition on wearable devices," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24471–24481, 2021.

[139] P. Mayer, M. Magno, and L. Benini, "Energy-positive activity recognition - from kinetic energy harvesting to smart self-sustainable wearable devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, no. 5, pp. 926–937, 2021.

[140] F. Samie, L. Bauer, and J. Henkel, "Hierarchical classification for constrained iot devices: A case study on human activity recognition," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8287–8295, 2020.

[141] F. Daghero, D. J. Pagliari, and M. Poncino, "Two-stage human activity recognition on microcontrollers with decision trees and cnns," in *2022 17th Conference on Ph. D Research in Microelectronics and Electronics (PRIME)*, pp. 173–176, IEEE, 2022.

[142] M. Odema, N. Rashid, and M. A. Al Faruque, "Eexnas: Early-exit neural architecture search solutions for low-power wearable devices," in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6, IEEE, 2021.

[143] N. Rashid, B. U. Demirel, and M. A. Al Faruque, "Ahar: Adaptive cnn for energy-efficient human activity recognition in low-power edge devices," *IEEE Internet of Things Journal*, 2022.

[144] N. Rashid, M. Dautta, P. Tseng, and M. A. Al Faruque, "Hear: Fog-enabled energy-aware online human eating activity recognition," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 860–868, 2021.

[145] N. Ellis, S. Cittolin, and L. Mapelli, "Signal processing, triggering and data acquisition," *New Technologies for Supercolliders*, pp. 361–385, 1991.

[146] L. Huang, M. Garofalakis, J. Hellerstein, A. Joseph, and N. Taft, "Toward sophisticated detection with distributed triggers," in *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pp. 311–316, 2006.

[147] K. Rana, R. Singh, and K. Sayann, "Correlation based novel technique for real time oscilloscope triggering for complex waveforms," *Measurement*, vol. 43, no. 3, pp. 299–311, 2010.

[148] A. Bogliolo, E. Lattanzi, and V. Freschi, "Idleness as a resource in energy-neutral wsns," in *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*, pp. 1–6, 2013.

[149] U. Raza, A. Bogliolo, V. Freschi, E. Lattanzi, and A. L. Murphy, "A two-prong approach to energy-efficient wsns: Wake-up receivers plus dedicated, model-based sensing," *Ad Hoc Networks*, vol. 45, pp. 1–12, 2016.

[150] E. Zaraket, N. M. Murad, S. S. Yazdani, L. Rajaoarisoa, and B. Ravelo, "An overview on low energy wake-up radio technology: Active and passive circuits associated with mac and routing protocols," *Journal of Network and Computer Applications*, vol. 190, p. 103140, 2021.

[151] O. Ullah, M. Hanan, and M. A. Ghafoor, "Energy efficiency issues in android application: A literature review," in *2022 24th International Multitopic Conference (INMIC)*, pp. 1–6, IEEE, 2022.

[152] F. Palomba, D. Di Nucci, A. Panichella, A. Zaidman, and A. De Lucia, "On the impact of code smells on the energy consumption of mobile applications," *Information and Software Technology*, vol. 105, pp. 43–55, 2019.

[153] H. Wu, H. Zhang, Y. Wang, and A. Rountev, "Sentinel: generating gui tests for sensor leaks in android and android wear apps," *Software Quality Journal*, vol. 28, pp. 335–367, 2020.

[154] L. Cruz and R. Abreu, "Catalog of energy patterns for mobile applications," *Empirical Software Engineering*, vol. 24, pp. 2209–2235, 2019.

[155] L. Corbalan, J. Fernandez, A. Cuitiño, L. Delia, G. Cáseres, P. Thomas, and P. Pesado, "Development frameworks for mobile devices: a comparative study about energy consumption," in *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems*, pp. 191–201, 2018.

[156] S. Huber, L. Demetz, and M. Felderer, "A comparative study on the energy consumption of progressive web apps," *Information Systems*, vol. 108, p. 102017, 2022.

[157] D. H. Gawali and V. M. Wadhai, "Recent trends in energy management of wireless wearable bio sensor design," in *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pp. 408–412, IEEE, 2017.

[158] V. Dissanayake, D. S. Elvitigala, H. Zhang, C. Weerasinghe, and S. Nanayakkara, "Comprate: Power efficient heart rate and heart rate variability monitoring on smart wearables," in *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–8, 2019.

[159] J. Ko, Y.-J. Choi, and R. Paul, "Computation offloading technique for energy efficiency of smart devices," *Journal of Cloud Computing*, vol. 10, no. 1, p. 44, 2021.

[160] H. Zhang, H. Wu, and A. Rountev, "Detection of energy inefficiencies in android wear watch faces," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 691–702, 2018.

[161] K. Martin, A. Wijekoon, and N. Wiratunga, "Human activity recognition with deep metric learners.," in *ICCBR Workshops*, CEUR Workshop Proceedings, 2020.

[162] M. Zeeshan, A. Pandey, and S. Kumar, "Csi-based device-free joint activity recognition and localization using siamese networks," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pp. 260–264, IEEE, 2022.

[163] S. Sani, N. Wiratunga, S. Massie, and K. Cooper, "Personalised human activity recognition using matching networks," in *Case-Based Reasoning Research and Development: 26th International Conference, ICCBR 2018, Stockholm, Sweden, July 9-12, 2018, Proceedings 26*, pp. 339–353, Springer, 2018.

[164] U. O. Matthew, R. L. Rosa, N. U. Okafor, J. S. Kazaure, O. D. Okey, M. A. Oladipupo, L. O. Fatai, V. N. Oriakhi, and D. Z. Rodriguez, "5g light-emitting biomedical robots for hospital disinfection," in *2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON)*, pp. 1–5, IEEE, 2024.

[165] R. K. Megalingam, A. Raj, S. K. Manoharan, and V. Egumadiri, "App based teleoperated uv disinfectant robot for covid cause," in *2021 Second International conference on electronics and sustainable communication systems (ICESC)*, pp. 277–281, IEEE, 2021.

[166] R. Bormann, F. Weisshardt, G. Arbeiter, and J. Fischer, "Autonomous dirt detection for cleaning in office environments," in *2013 IEEE international conference on robotics and automation*, pp. 1260–1267, IEEE, 2013.

[167] K. Ruan, Z. Wu, and Q. Xu, "Smart cleaner: A new autonomous indoor disinfection robot for combating the covid-19 pandemic," *Robotics*, vol. 10, no. 3, p. 87, 2021.

[168] X. Feng, P. Hu, T. Jin, J. Fang, F. Tang, H. Jiang, and C. Lu, "On-site monitoring of airborne pathogens: recent advances in bioaerosol collection and rapid detection," *Aerobiologia*, vol. 40, no. 3, p. 303–341, 2024.

[169] V. R, A. S, G. N, S. N. K. S, and S. E. A. M, "Smart iot enabled cleaner with lidar navigation and auto-docking," *International Journal of Engineering Research and Sustainable Technologies (IJERST)*, vol. 3, no. 1, p. 22–29, 2025.

[170] H. Ji and H. Huang, "The integration and development trend of china's 5g technology and smart cleaning," *Journal of Physics: Conference Series*, vol. 1812, no. 1, p. 012015, 2021.

[171] M. Mantelli, L. Dos Santos, L. de Fraga, G. Miotto, A. Bergamin, E. Cardoso, M. Serrano, R. Maffei, E. Prestes, J. Netto, *et al.*, "Autonomous environment disinfection based on dynamic uv-c irradiation map," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4789–4796, 2022.

[172] P. Mahajan, N. Ponde, A. Malvi, A. Gupta, and D. Deshmukh, "Design and development of smart home cleaning robot," *Research gate*, 2021.

[173] S. P. Yadav, S. Kumar, B. Devika, and K. Rahul, "Robot-assisted ultraviolet disinfector with dispenser for healthcare related services," *European Journal of Electrical Engineering and Computer Science*, vol. 6, no. 1, pp. 1–5, 2022.

[174] P. Tripathi, A. Pradhan, S. Khandelwal, A. Singh, and A. Singh, "Work authentication and monitoring system with blockchain and internet of things (iot)," *Advances and Applications in Mathematical Sciences*, vol. 21, no. 5, pp. 2661–2680, 2022.

[175] P. Choden, T. Seesaard, U. Dorji, C. Sriphrapradang, and T. Kerdcharoen, "Urine odor detection by electronic nose for smart toilet application," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 190–193, IEEE, 2017.

[176] R. Yatabe, A. Shunori, B. Wyszynski, Y. Hanai, A. Nakao, M. Nakatani, A. Oki, H. Oka, T. Washio, and K. Toko, "Odor sensor system using chemosensitive resistor array and machine learning," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 2077–2083, 2020.

[177] C. A. Cid, E. Shafran, S. I. Jellal, J. Field, R. Le Floch, C. Paules, Y. El Hilali, M. R. Hoffmann, *et al.*, "Self-diagnosis and smart maintenance prototype for sustainable and desirable onsite sanitation," in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–4, IEEE, 2020.

[178] P. Namekar and B. Karthikeyan, *Integration of the Smart Phone and IOT for Smart Public Toilet Hygiene Monitoring System*, p. 77–82. Springer Singapore, 2018.

[179] K. Boonyakan, N. Heamra, and A. Changkamanon, "Water efficient toilet: Setting a suitable automatic flushing duration," in *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*, pp. 143–146, IEEE, 2018.

[180] Y. Abhishek, K. Bharath, A. Jayanth Kumar, J. Nanditha, and S. B. Dr. Manoj Kumar, "Iot enabled smart washroom," *International Journal of Innovative Research in Engineering*, vol. 5, no. 3, pp. 156–163, 2024.

[181] I. Thennakoon, P. Hewawasam, D. Wijesundara, N. Fernando, L. Gunawardena, and C. Premachandra, "A framework for iot-enabled smart washrooms," in *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*, pp. 612–613, 2021.

[182] A. Waje, N. Shaikh, P. Pansare, and P. Kamble, "Toilet hygiene system," in *In Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)*, 2020.

[183] D. Siroya, S. Prusty, P. Shah, M. Kavedia, and A. Hatekar, "Iot based washroom feedback system for quality monitoring," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 10, no. 4, 2022.

[184] C. P. Choo and J. Jalaludin, "An overview of indoor air quality and its impact on respiratory health among malaysian school-aged children," *Reviews on Environmental Health*, vol. 30, no. 1, pp. 9–18, 2015.

[185] C.-C. Chen, L.-H. Chen, and M.-H. Guo, "Integration of building interface and smart sensor network to control indoor air pollution through internet of things," in *Proceedings of the 8th International Conference on Informatics, Environment, Energy and Applications*, pp. 15–20, 2019.

[186] Z. Liu, G. Wang, L. Zhao, and G. Yang, "Multi-points indoor air quality monitoring based on internet of things," *IEEE access*, vol. 9, pp. 70479–70492, 2021.

[187] X. Dai, W. Shang, J. Liu, M. Xue, and C. Wang, "Achieving better indoor air quality with iot systems for future buildings: Opportunities and challenges," *Science of The Total Environment*, vol. 895, pp. 1–14, 2023.

[188] Q. P. Ha, S. Metia, and M. D. Phung, "Sensing data fusion for enhanced indoor air quality monitoring," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 4430–4441, 2020.

[189] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–35, 2016.

[190] A. S. Shah, H. Nasir, M. Fayaz, A. Lajis, and A. Shah, "A review on energy consumption optimization techniques in iot based smart building environments," *Information*, vol. 10, no. 3, p. 108, 2019.

[191] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "What does model-driven data acquisition really achieve in wireless sensor networks?," in *2012 IEEE International Conference on Pervasive Computing and Communications*, pp. 85–94, IEEE, 2012.

[192] H. Harb, C. A. Jaoude, and A. Makhoul, "An energy-efficient data prediction and processing approach for the internet of things and sensing based applications," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 780–795, 2020.

[193] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3816–3825, 2013.

[194] F. A. Aderohunmu, G. Paci, D. Brunelli, J. D. Deng, L. Benini, and M. Purvis, "An application-specific forecasting algorithm for extending wsn lifetime," in *2013 IEEE international conference on distributed computing in sensor systems*, pp. 374–381, IEEE, 2013.

[195] T. Pötsch, L. Pei, K. Kuladinithi, and C. Goerg, "Model-driven data acquisition for temperature sensor readings in wireless sensor networks," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6, IEEE, 2014.

[196] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-based approximate querying in sensor networks," *The VLDB journal*, vol. 14, pp. 417–443, 2005.

[197] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 588–599, 2004.

[198] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *22nd International Conference on Data Engineering (ICDE'06)*, pp. 48–48, IEEE, 2006.

[199] I. Lazaridis and S. Mehrotra, "Capturing sensor-generated time series with quality guarantees," in *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*, pp. 429–440, IEEE, 2003.

[200] D. Tulone and S. Madden, "Paq: Time series forecasting for approximate query answering in sensor networks," in *European Workshop on Wireless Sensor Networks*, pp. 21–37, Springer, 2006.

[201] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pp. 191–300, 2006.

[202] A. Bogliolo, V. Freschi, E. Lattanzi, A. L. Murphy, and U. Raza, "Towards a true energetically sustainable wsn: A case study with prediction-based data collection and a wake-up receiver," in *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, pp. 21–28, IEEE, 2014.

[203] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[204] A. Jain, E. Y. Chang, and Y.-F. Wang, "Adaptive stream resource management using kalman filters," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 11–22, 2004.

[205] B. Gedik, L. Liu, and S. Y. Philip, "Asap: An adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and distributed systems*, vol. 18, no. 12, pp. 1766–1783, 2007.

[206] M. Giordano, S. Cortesi, P.-V. Mekikis, M. Crabolu, G. Bellusci, and M. Magno, "Energy-aware adaptive sampling for self-sustainability in resource-constrained iot devices," in *Proceedings of the 11th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, pp. 65–71, 2023.

[207] Y. Ben-Aboud, D. B. Licea, M. Ghogho, and A. Kobbane, "On adaptive sampling algorithms for iot devices," in *ICC 2021-IEEE International Conference on Communications*, pp. 1–7, IEEE, 2021.

[208] Y. W. Law, S. Chatterjea, J. Jin, T. Hanselmann, and M. Palaniswami, "Energy-efficient data acquisition by adaptive sampling for wireless sensor networks," in *Proceedings of the 2009 international conference on wireless communications and mobile computing: Connecting the world wirelessly*, pp. 1146–1151, 2009.

[209] W. Lalouani, M. Younis, I. White-Gittens, R. N. Emokpae Jr, and L. E. Emokpae, "Energy-efficient collection of wearable sensor data through predictive sampling," *Smart Health*, vol. 21, p. 100208, 2021.

[210] J.-F. Yang, Y.-J. Zhai, D.-P. Xu, and P. Han, "Smo algorithm applied in time series model building and forecast," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2395–2400, IEEE, 2007.

[211] A. Lahouar and J. B. H. Slama, "Day-ahead load forecast using random forest and expert input selection," *Energy Conversion and Management*, vol. 103, pp. 1040–1051, 2015.

[212] J. Han, G. H. Lee, S. Park, J. Lee, and J. K. Choi, "A multivariate-time-series-prediction-based adaptive data transmission period control algorithm for iot networks," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 419–436, 2021.

[213] C. Cao, J. Huang, M. Wu, Z. Lin, and Y. Sun, "A multivariate time series prediction method based on convolution-residual gated recurrent neural network and double-layer attention," *Electronics*, vol. 13, no. 14, p. 2834, 2024.

[214] İ. Kök and S. Özdemir, "Deepmdp: A novel deep-learning-based missing data prediction protocol for iot," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 232–243, 2020.

[215] J. He, Y. Li, X. Zhang, and J. Li, "Missing and corrupted data recovery in wireless sensor networks based on weighted robust principal component analysis," *Sensors*, vol. 22, no. 5, p. 1992, 2022.

[216] N. A. Khan and S. Ali, "Robust sparse reconstruction of signals with gapped missing samples from multi-sensor recordings," *Digital Signal Processing*, vol. 123, p. 103392, 2022.

[217] N. Fatima, S. Riaz, S. Ali, R. Khan, M. Ullah, and D. Kwak, "Sensors faults classification and faulty signals reconstruction using deep learning," *IEEE Access*, 2024.

[218] A. M. Rahmani, J. Tanveer, F. S. Gharehchopogh, S. Rajabi, and M. Hosseinzadeh, "A novel offloading strategy for multi-user optimization in blockchain-enabled mobile edge computing networks for improved internet of things performance," *Computers and Electrical Engineering*, vol. 119, p. 109514, 2024.

[219] K. Moghaddasi, S. Rajabi, F. S. Gharehchopogh, and A. Ghaffari, "An advanced deep reinforcement learning algorithm for three-layer d2d-edge-cloud computing architecture for efficient task offloading in the internet of things," *Sustainable Computing: Informatics and Systems*, vol. 43, p. 100992, 2024.

[220] P. Capellacci, L. Calisti, and E. Lattanzi, "A low-cost iot sensor for indoor monitoring with prediction-based data collection.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 11, 2024.

[221] B. Płaczek, "Prediction-based data reduction with dynamic target node selection in iot sensor networks," *Future Generation Computer Systems*, vol. 152, pp. 225–238, 2024.

[222] Q. Han, S. Mehrotra, and N. Venkatasubramanian, "Energy efficient data collection in distributed sensor environments," in *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, pp. 590–597, IEEE, 2004.

[223] A. M. Hussein, A. K. Idrees, and R. Couturier, "A distributed prediction–compression-based mechanism for energy saving in iot networks," *The Journal of Supercomputing*, vol. 79, no. 15, pp. 16963–16999, 2023.

[224] S. Pandey, K. Dubey, R. Dubey, and S. Kumar, "Eedcs: Energy efficient data collection schemes for iot enabled wireless sensor network," *Wireless Personal Communications*, vol. 129, no. 2, pp. 1297–1313, 2023.

[225] M. Chandradhara, A. George, M. Faraaz, and A. Saraf, "Design of iot based smart home system," *Journal of University of Shanghai for Science and Technology*, vol. 23, no. 12, p. 249–261, 2021.

[226] M. De Donno, K. P. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog," *IEEE access*, vol. 7, pp. 150936–150948, 2019.

[227] A. Al-Dulaimy, M. Jansen, B. Johansson, A. Trivedi, A. Iosup, M. Ashjaei, A. Galletta, D. Kimovski, R. Prodan, K. Tserpes, *et al.*, "The computing continuum: From iot to the cloud," *Internet of Things*, vol. 27, p. 101272, 2024.

[228] S. S. Goel, A. Goel, M. Kumar, and G. Moltó, "A review of internet of things: qualifying technologies and boundless horizon," *Journal of Reliable Intelligent Environments*, vol. 7, no. 1, pp. 23–33, 2021.

[229] E. Lattanzi, M. Dromedari, and V. Freschi, "A scalable multitasking wireless sensor network testbed for monitoring indoor human comfort," *IEEE Access*, vol. 6, pp. 17952–17967, 2018.

[230] J. Saini, M. Dutta, and G. Marques, "Indoor air quality monitoring systems based on internet of things: A systematic review," *International journal of environmental research and public health*, vol. 17, no. 14, p. 4942, 2020.

[231] L. Bianconi, Y. Lechiara, L. Bixio, R. Palermo, S. Pensieri, F. Viti, and R. Bozzano, "Edge and fog computing for iot: A case study for citizen well-being," in *International Summit Smart City 360°*, pp. 121–139, Springer, 2021.

[232] T. L. Narayana, C. Venkatesh, A. Kiran, A. Kumar, S. B. Khan, A. Almusharraf, M. T. Quasim, *et al.*, "Advances in real time smart monitoring of environmental parameters using iot and sensors," *Heliyon*, vol. 10, no. 7, 2024.

[233] S. Sonawani and K. Patil, "Air quality measurement, prediction and warning using transfer learning based iot system for ambient assisted living," *International Journal of Pervasive Computing and Communications*, vol. 20, no. 1, pp. 38–55, 2024.

[234] W.-T. Sung and S.-J. Hsiao, "Building an indoor air quality monitoring system based on the architecture of the internet of things," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, pp. 1–41, 2021.

[235] Y. Zhu, S. A. Al-Ahmed, M. Z. Shakir, and J. I. Olszewska, "Lstm-based iot-enabled co2 steady-state forecasting for indoor air quality monitoring," *Electronics*, vol. 12, no. 1, p. 107, 2022.

[236] H. K. Apat, R. Nayak, and B. Sahoo, "A comprehensive review on internet of things application placement in fog computing environment," *Internet of Things*, p. 100866, 2023.

[237] J. Wei, Y. Wang, J. Mo, and C. Fan, "One-year dataset of hourly air quality parameters from 100 air purifiers used in china residential buildings," *Scientific Data*, vol. 10, no. 1, 2023.

[238] L. Sun, P. Wei, D. Westerdahl, J. Xue, and Z. Ning, "Evaluating indoor air quality in schools: Is the indoor environment a haven during high pollution episodes?," *Toxics*, vol. 12, no. 8, p. 564, 2024.

[239] P. Karmakar, S. Pradhan, and S. Chakraborty, "Indoor air quality dataset with activities of daily living in low to middle-income communities," 2024.

[240] A. Rejeb, K. Rejeb, S. Simske, H. Treiblmaier, and S. Zailani, "The big picture on the internet of things and the smart city: a review of what we know and what we need to know," *Internet of Things*, vol. 19, p. 100565, 2022.

[241] E. H. Houssein, M. A. Othman, W. M. Mohamed, and M. Younan, "Internet of things in smart cities: Comprehensive review, open issues and challenges," *IEEE Internet of Things Journal*, 2024.

[242] G. Paolone, D. Iachetti, R. Paesani, F. Pilotti, M. Marinelli, and P. Di Felice, "A holistic overview of the internet of things ecosystem," *IoT*, vol. 3, no. 4, pp. 398–434, 2022.

[243] M. Medojević, N. Marković, and A. Rikalović, "Modeling ai-driven iot energy consumption: From device-level forecasts to system-level dynamics," in *2025 10th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–7, IEEE, 2025.

[244] N. Jain and M. Mandot, "Energy efficient strategies in internet of things: an overview," *ICT for Competitive Strategies*, pp. 585–592, 2020.

[245] E. Hittinger and P. Jaramillo, "Internet of things: energy boon or bane?," *Science*, vol. 364, no. 6438, pp. 326–328, 2019.

[246] E. A. de Oliveira, F. C. Delicato, A. R. da Rocha, and M. Mattoso, "A real-time and energy-aware framework for data stream processing in the internet of things.," in *IoTBDS*, pp. 17–28, 2021.

[247] Y. L. Cheng, M. H. Lim, and K. H. Hui, "Impact of internet of things paradigm towards energy consumption prediction: A systematic literature review," *Sustainable Cities and Society*, vol. 78, p. 103624, 2022.

[248] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[249] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[250] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge computing with artificial intelligence: A machine learning perspective," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[251] H. Djelouat, A. Amira, and F. Bensaali, "Compressive sensing-based iot applications: A review," *Journal of Sensor and Actuator Networks*, vol. 7, no. 4, p. 45, 2018.

[252] H. A. Selmy, H. K. Mohamed, and W. Medhat, "A predictive analytics framework for sensor data using time series and deep learning techniques," *Neural Computing and Applications*, vol. 36, no. 11, pp. 6119–6132, 2024.

[253] M. A. Ala'anzy, R. Zhanuzak, R. Akhmedov, N. Mohamed, and J. Al-Jaroodi, "Dynamic load balancing for enhanced network performance in iot-enabled smart healthcare with fog computing," *IEEE Access*, 2024.

[254] A. Bourechak, O. Zedadra, M. N. Kouahla, A. Guerrieri, H. Seridi, and G. Fortino, "At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives," *Sensors*, vol. 23, no. 3, p. 1639, 2023.

[255] S.-m. Park, D. D. Won, B. J. Lee, D. Escobedo, A. Esteva, A. Aalipour, T. J. Ge, J. H. Kim, S. Suh, E. H. Choi, A. X. Lozano, C. Yao, S. Bodapati, F. B. Achterberg, J. Kim, H. Park, Y. Choi, W. J. Kim, J. H. Yu, A. M. Bhatt, J. K. Lee, R. Spitler, S. X. Wang, and S. S. Gambhir, "A mountable toilet system for personalized health monitoring via the analysis of excreta," *Nature Biomedical Engineering*, vol. 4, no. 6, p. 624–635, 2020.

[256] F. Svoboda, J. Fernandez-Marques, E. Liberis, and N. D. Lane, "Deep learning on microcontrollers: A study on deployment costs and challenges," in *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, pp. 54–63, 2022.

[257] Espressif Systems, "ESP-IDF — ESP IoT Development Framework." `https://www.espressif.com/en/products/sdks/esp-idf`, 2025. Online; Accessed 26-August-2025.

[258] FreeRTOS, "Freertos — real-time operating system for microcontrollers." `https://www.freertos.org`, 2024. Online; Accessed 26-August-2025.

[259] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.

[260] Winsen Sensor, "MH-Z19B — Infrared CO2 Sensor Module." https://www.winsen-sensor.com/d/files/MH-Z19B.pdf, 2025. Online; Accessed 26-August-2025.

[261] Sensirion, the Sensor Company, "Datasheet SGP30 — Indoor Air Quality Sensor for TVOC and CO2 Measurements." https://sensirion.com/media/documents/984E0DD5/61644B8B/Sensirion_Gas_Sensors_Datasheet_SGP30.pdf, 2025. Online; Accessed 26-August-2025.

[262] Mouser Electronics, "DHT11 — Humidity & Temperature Sensor." https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf?srsltid=AfmBOor3kriOVkWBjOm-xO4JqSVXzddtK9-1Jha1f_EG3EipSTsWMgT6, 2025. Online; Accessed 26-August-2025.

[263] Sensirion, the Sensor Company, "Datasheet SPS30 — Particulate Matter Sensor for Air Quality Monitoring and Control." https://cdn.sparkfun.com/assets/2/d/2/a/6/Sensirion_SPS30_Particulate_Matter_Sensor_v0.9_D1__1_.pdf, 2025. Online; Accessed 26-August-2025.

[264] InvenSense, "INMP411 — Omnidirectional Microphone with Bottom Port and I2S Digital Output." https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf, 2025. Online; Accessed 05-November-2025.

[265] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2231–2244, 2015.

[266] F. Chollet *et al.*, "Keras." https://keras.io, 2015. Online; Accessed 27-August-2025.

[267] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems." https://www.tensorflow.org/, 2015. Online; Accessed 27-August-2025.

[268] Google AI Edge, "LiteRT Overview — Google AI for Developers." https://ai.google.dev/edge/litert, 2025. Online; Accessed 27-August-2025.

[269] Google AI Edge, "LiteRT for Microcontroller — Google AI for Developers." https://ai.google.dev/edge/litert/microcontrollers, 2025. Online; Accessed 27-August-2025.

[270] Espressif Systems, "ESP32 — Technical Documents." https://www.espressif.com/en/support/documents/technical-documents, 2025. Online; Accessed 26-August-2025.

[271] Raspberry Pi, "Raspberry Pi Pico 2 W." https://www.raspberrypi.com/documentation/microcontrollers/pico-series.html#pico2w-technical-specification, 2025. Online; Accessed 07-December-2025.

[272] Raspberry Pi, "The C/C++ SDK." https://www.raspberrypi.com/documentation/microcontrollers/c_sdk.html, 2025. Online; Accessed 07-December-2025.

[273] Raspberry Pi, "MicroPython." https://www.raspberrypi.com/documentation/microcontrollers/micropython.html, 2025. Online; Accessed 07-December-2025.

[274] NXP, "MCXN947 datasheet." https://www.nxp.com/docs/en/data-sheet/MCXNx4xDS.pdf, 2025. Online; Accessed 07-December-2025.

[275] NXP, "MCUXpresso Integrated Development Environment (IDE)." https://www.nxp.com/design/design-center/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE, 2025. Online; Accessed 07-December-2025.

[276] Rohde & Schwarz, "Dual-Channel Analyzer/Power Supply NGMO2." https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/ngmo2_21_web-LF.pdf, 2025. Online; Accessed 05-September-2025.

[277] National Instrument, "PCI/PCIe/PXI/PXIe/USB 6251 Specifications." https://www.ni.com/docs/en-US/bundle/pci-pcie-pxi-pxie-usb-6251-specs/page/specs.html, 2025. Online; Accessed 05-September-2025.

[278] National Instrument, "BNC-2120 Installation Guide." https://www.ni.com/docs/en-US/bundle/bnc-2120-getting-started/resource/372123d.pdf, 2025. Online; Accessed 05-September-2025.

[279] Espressif, "ESP-NN: The library contains optimised NN (Neural Network) functions for various Espressif chips." https://github.com/espressif/esp-nn, 2025. Online; Accessed 10-July-2025.

[280] D. F. Orozco Lopez, *Study the feasibility of a system that detects the presence of people in hotel rooms to save energy when the rooms are vacated.* PhD thesis, Politecnico di Torino, 2024.

[281] P. Chaudhari, Y. Xiao, M. M.-C. Cheng, and T. Li, "Fundamentals, algorithms, and technologies of occupancy detection for smart buildings using iot sensors," *Sensors*, vol. 24, no. 7, p. 2123, 2024.

[282] M. L, U. E, P. K R, and R. K S, "Smart walking cane for visually impaired individuals using stm32h563zi and time-of-flight sensors with haptic feedback (sense stride)," *ITM Web of Conferences*, vol. 79, p. 01002, 2025.

[283] M. Toa and A. Whitehead, "Ultrasonic sensing basics," *Dallas: Texas Instruments*, pp. 53–75, 2020.

[284] S. V, P. D, S. N, S. S, and V. S, "Smart occupancy detection and activity recognition using rf transmissions," in *2025 International Conference on Computing and Communication Technologies (ICCCT)*, p. 1–5, IEEE, 2025.

[285] A. N. Mohamed, "A novice guide towards human motion analysis and understanding," *arXiv preprint arXiv:1509.01074*, 2015.

[286] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[287] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[288] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR, abs/1506.02640*, 2015.

[289] R. Sapkota, M. Flores-Calero, R. Qureshi, C. Badgujar, U. Nepal, A. Poulose, P. Zeno, U. B. P. Vaddevolu, S. Khan, M. Shoman, *et al.*, "Yolo advances to its genesis: A decadal and comprehensive review of the you only look once (yolo) series," *Artificial Intelligence Review*, vol. 58, no. 9, p. 274, 2025.

[290] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "Mobilenetv2 model for image classification," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pp. 476–480, IEEE, 2020.

[291] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen, and T.-T. Lee, "Mobilenet-ssdv2: An improved object detection model for embedded systems," in *2020 International conference on system science and engineering (ICSSE)*, pp. 1–5, IEEE, 2020.

[292] NVIDIA, "Trafficcamnet." `https://catalog.ngc.nvidia.com/orgs/nvidia/teams/tao/models/trafficcamnet`, 2023. Online; Accessed 10-October-2023.

[293] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[294] J. Gong, W. Liu, M. Pei, C. Wu, and L. Guo, "Resnet10: A lightweight residual network for remote sensing image classification," in *2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 975–978, IEEE, 2022.

[295] H. Luo and Z. Zeng, "Real-time multi-object tracking based on bi-directional matching," *arXiv preprint arXiv:2303.08444*, 2023.

[296] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, IEEE, 2016.

[297] T. Azfar, C. Wang, R. Ke, A. Raheem, J. Weidner, and R. L. Cheu, "Incorporating vehicle detection algorithms via edge computing on a campus digital twin model," in *International Conference on Transportation and Development 2023*, pp. 400–409, ASCE, 2023.

[298] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[299] A. H. Jonathon Luiten, "Trackeval." `https://github.com/JonathonLuiten/TrackEval`, 2020. Online; Accessed 21-January-2026.

[300] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, no. 2, pp. 548–578, 2021.

[301] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, p. 246309, 2008.

[302] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*, pp. 17–35, Springer, 2016.

[303] G. Coral, "Coral edgetpu compiler." `https://coral.ai/docs/edgetpu/compiler/#system-requirements`, 2023. Online; Accessed 17-December-2023.

[304] Google, "Pycoral api." `https://coral.ai/docs/reference/py/`, 2020. Online; Accessed 28-September-2023.

[305] NVIDIA, "Deepstream." `https://developer.nvidia.com/deepstream-sdk`, 2023. Online; Accessed 01-October-2023.

[306] dusty nv, "Jetson inference." `https://github.com/dusty-nv/jetson-inference`, 2017. Online; Accessed 03-October-2023.

[307] NVIDIA, "Jetson orin for next-gen robotics | nvidia." `https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/`, 2023. Online; Accessed 11-December-2023.

[308] Google, "Dev board | coral." `https://coral.ai/products/dev-board/`, 2023. Online; Accessed 11-December-2023.

[309] Life, "Stillcamlife - 16 minutes in budapest hungary beautiful people walking and cars driving by." `https://www.youtube.com/watch?v=N79f1znMWQ8`, 2014. Online; Accessed 11-December-2023.

[310] TexasHighDef, "Cars driving on route 28 "road noise"." `https://www.youtube.com/watch?v=FOo0AbigryE`, 2019. Online; Accessed 11-December-2023.

[311] T. Chaudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Borriello, B. Hemingway, *et al.*, "The mobile sensing platform: an embedded system for activity recognition," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 32–41, 2008.

[312] P. P. Ariza-Colpas, E. Vicario, A. I. Oviedo-Carrascal, S. Butt Aziz, M. A. Piñeres-Melo, A. Quintero-Linero, and F. Patara, "Human activity recognition data analysis: History, evolutions, and new trends," *Sensors*, vol. 22, no. 9, p. 3401, 2022.

[313] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, p. 91–102, 1992.

[314] E. Charniak and R. P. Goldman, "A bayesian model of plan recognition," *Artificial Intelligence*, vol. 64, no. 1, p. 53–79, 1993.

[315] D.-A. Nguyen and N.-A. Le-Khac, "Sok: behind the accuracy of complex human activity recognition using deep learning," in *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2024.

[316] S. Majumder and N. Kehtarnavaz, "Vision and inertial sensing fusion for human action recognition: A review," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 2454–2467, 2020.

[317] F. Salice, M. De Vellis, A. G. Barbieri, A. Masciadri, and S. Comai, "Nonintrusive monitoring and detection of sitting and lying persons: A technological review," *IEEE Access*, 2024.

[318] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[319] Y. Cai, B. Guo, F. Salim, and Z. Hong, "Towards generalizable human activity recognition: A survey," *arXiv preprint arXiv:2508.12213*, 2025.

[320] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[321] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," *International conference on pervasive computing*, pp. 319–336, 2010.

[322] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[323] C. Hou, "A study on IMU-based human activity recognition using deep learning and traditional machine learning," *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, pp. 225–234, 2020.

[324] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[325] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Classifier and exemplar synthesis for zero-shot learning," *International Journal of Computer Vision*, vol. 128, pp. 166–201, 2020.

[326] A. M. Research, "Smartwatch market by product (extension, standalone, and classical), application (personal assistance, wellness, healthcare, sports, and others), and operating system (watchos, android, rtos, tizen, and others): Global opportunity analysis and industry forecast, 2020-2027." Online; Accessed 26-April-2023.

[327] S. M. Mishra, *Wearable android: android wear and google fit app development*. John Wiley & Sons, 2015.

[328] F. M. Talaat and R. M. El-Balka, "Stress monitoring using wearable sensors: Iot techniques in medical field," *Neural Computing and Applications*, vol. 35, no. 25, pp. 18571–18584, 2023.

[329] E. Lattanzi and L. Calisti, "Energy-aware tiny machine learning for sensor-based hand-washing recognition," in *Proceedings of the 2023 8th International Conference on Machine Learning Technologies*, pp. 15–22, 2023.

[330] P. Basu and J. Redi, "Effect of overhearing transmissions on energy efficiency in dense sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 196–204, 2004.

[331] J. Kim, S. Kim, J. Yun, and Y. Won, "Energy efficient io stack design for wearable device," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 2152–2159, 2019.

[332] L. Calisti and E. Lattanzi, "Sensorlib: an energy-efficient sensor-collection library for wear os," in *Proceedings of the 4th European Symposium on Software Engineering*, pp. 83–88, 2023.

[333] L. Calisti and E. Lattanzi, "Real-time energy-efficient sensor libraries for wearable devices," *IEEE Access*, 2024.

[334] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[335] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*, pp. 1243–1252, PMLR, 2017.

[336] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[337] Y. Shavit and I. Klein, "Boosting inertial-based human activity recognition with transformers," *IEEE Access*, vol. 9, pp. 53540–53547, 2021.

[338] I. Dirgová Luptáková, M. Kubovčík, and J. Pospíchal, "Wearable sensor-based human activity recognition with transformer model," *Sensors*, vol. 22, no. 5, p. 1911, 2022.

[339] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[340] M. Jacobsen and M. Tropmann-Frick, "Imputation strategies in time series based on language models," *Datenbank-Spektrum*, pp. 1–11, 2024.

[341] A. Crivellari, B. Resch, and Y. Shi, "Tracebert—a feasibility study on reconstructing spatial–temporal gaps from incomplete motion trajectories via bert training process on discrete location sequences," *Sensors*, vol. 22, no. 4, p. 1682, 2022.

[342] M. Alzantot, S. Chakraborty, and M. Srivastava, "Sensegen: A deep learning architecture for synthetic sensor data generation," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 188–193, IEEE, 2017.

[343] W. You, H. Jiang, Z. Liu, Z. Xie, T. Liu, J. Lu, and F. Dou, "Adlgen: Synthesizing symbolic, event-triggered sensor sequences for human activity modeling," *arXiv preprint arXiv:2505.17987*, 2025.

[344] A. Hussain, S. U. Khan, N. Khan, M. W. Bhatt, A. Farouk, J. Bhola, and S. W. Baik, "A hybrid transformer framework for efficient activity recognition using consumer electronics," *IEEE Transactions on Consumer Electronics*, 2024.

[345] F. Bianco Morghet, *Application of Transformers to edge-computing in ultra-low power devices*. PhD thesis, Politecnico di Torino, 2021.

[346] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.

[347] C.-Y. Lin and R. Marculescu, "Model personalization for human activity recognition," in *2020 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops)*, pp. 1–7, IEEE, 2020.

[348] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[349] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[350] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "Attnsense: Multi-level attention mechanism for multimodal human activity recognition.," *IJCAI*, pp. 3109–3115, 2019.

[351] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, vol. 18, no. 185, pp. 1–52, 2018.

[352] C. Contoli and E. Lattanzi, "A study on the application of tensorflow compression techniques to human activity recognition," *IEEE Access*, 2023.

[353] L. Bigelli, C. Contoli, V. Freschi, and E. Lattanzi, "Privacy preservation in sensor-based human activity recognition through autoencoders for low-power iot devices," *Internet of Things*, vol. 26, p. 101189, 2024.

[354] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, (New York, NY, USA), pp. 49–58, ACM, 2019.

[355] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones," in *International conference on information and communication technologies for ageing well and e-health*, vol. 2, pp. 143–151, SciTePress, 2016.

[356] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, IEEE, 2016.

[357] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (KDD)*, pp. 10–18, 2010.

[358] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," *21th European Symposium on Artificial Neural Networks (ESANN)*, 2013.

[359] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers (ISWC)*, pp. 108–109, IEEE, 2012.

[360] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[361] W. Wang and Q. Li, "Generalized zero-shot activity recognition with embedding-based method," *ACM Transactions on Sensor Networks*, vol. 19, no. 3, pp. 1–25, 2023.

[362] S. Nooruddin, M. M. Islam, and F. Karray, "Tinyhar: benchmarking human activity recognition systems in resource constrained devices," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pp. 1–8, IEEE, 2022.

[363] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *European Conference on Computer Vision–ECCV*, pp. 52–68, Springer, 2016.

[364] S. Ek, F. Portet, and P. Lalanda, "Transformer-based models to deal with heterogeneous environments in human activity recognition," *Personal and Ubiquitous Computing*, vol. 27, no. 6, pp. 2267–2280, 2023.