A Machine Learning Enabled Hall-Effect IoT-System for Monitoring Building Vibrations

Emanuele Lattanzi, Paolo Capellacci, Valerio Freschi
Dep. of Pure and Applied Sciences, University of Urbino,
Italy

Abstract—**Vibration monitoring of civil infrastructures is a fundamental task to assess their structural health, which can be nowadays carried on at reduced costs thanks to new sensing devices and embedded hardware platforms. In this work, we present a system for monitoring vibrations in buildings based on a novel, cheap, Hall-effect vibration sensor that is interfaced with a commercially available embedded hardware platform, in order to support communication toward cloud based services by means of IoT communication protocols. Two deep learning neural networks have been implemented and tested to demonstrate the capability of performing nontrivial prediction tasks directly on board of the embedded platform, an important feature to conceive dynamical policies for deciding whether to perform a recognition task on the final (resource constrained) device, or delegate it to the cloud according to specific energy, latency, accuracy requirements. Experimental evaluation on two use cases, namely the detection of a seismic event and the count of steps made by people transiting in a public building highlight the potential of the adopted solution; for instance, recognition of walking-induced vibrations can be achieved with an accuracy of 96% in real-time within time windows of 500ms. Overall, the results of the empirical investigation show the flexibility of the proposed solution as a promising alternative for the design of vibration monitoring systems in built environments.**

Keywords*—*Vibration sensor; machine learning; hall-effect

I. Introduction

The growing attention toward the structural efficiency of civil structures and infrastructures prompts the need for the design of novel and effective systems capable of monitoring structural health at adequate time and space resolution, and providing effective evaluation and support for downstream decision-making in order to mitigate risks, increase safety, reduce maintenance costs [1], [2].

Structural Health Monitoring (SHM), i.e. the set of techniques, methodologies, and technologies that enable to obtain information regarding the state of a structure (for instance in terms of the functioning of structures or their response), is therefore becoming a crucial aspect of the lifetime cycle of the built environment.

The analysis of the vibrational response of a building, encoded into a time series of displacements and/or of accelerations is a commonly adopted approach for SHM, for instance for establishing potential damage states. This response is usually the result of the non-trivial interplay between several physical properties of the structure (e.g. mass distribution, stiffness, damage pattern, damping sources) and operating conditions [3].

Indeed, SHM methods make use of the vibration response of the structure to derive conclusions regarding its overall conditions (e.g. healthy or damaged). This is generally achieved by means of the acceleration signals gathered from a network of sensors positioned at specific points of the structure, and by subsequent processing to extract from acceleration signals information regarding the possible presence of damage (and also its localization and degree of severity).

In general, SHM can nowadays benefit from a suite of technologies from different fields that can be exploited by designers to develop effective systems at reduced costs. Indeed, recent advancements in the development of novel sensor devices, the advent of various communication and computation methodologies to support the so called Internet of Things (IoT), and the successful application of sophisticated machine learning tools represent three enabling factors for building successful SHM systems.

In the following, we provide a brief summary of the main contributions and key aspects regarding the progress of each of the analyzed technologies.

A. Sensor Devices

Regarding the measurements of vibrations, accelerometers are commonly employed: they are in fact force-sensors coupled with a mass that applies a specific force proportional to the acceleration of the mass, according to the second Newton's law, when it is subject to vibration. Three types of transducers are mainly used to convert acceleration into proper electrical signals: piezoelectric, piezoresistive, and differential capacitive [1]. Piezoelectric accelerometers exploit the piezoelectric effect to measure changes occurring in specific materials undergoing mechanical stress. Piezoresistive sensors leverage the modifications induced in the electrical resistance of certain materials subject to mechanical deformation. Differential capacitive accelerometers measure changes in electrical capacitance to derive information about the displacement.

Regardless of the type of sensor, a common trend in many applications is the adoption of Micro-Electro-Mechanical-Systems (MEMS) as basic technology for SHM systems. Indeed, MEMS devices can be considered to be often on par with the performance of many macro-scale competitors, while they represent a lower cost and less invasive solution in many deployments.

B. Processing and Communication Infrastructures

For what concerns communication infrastructures, SHM solutions have traditionally been based on wired networks.

However, the increased reliability of several wireless sensor networks (WSN) core technologies, integrated within a general Internet of Things communication protocols framework, have recently emerged as competitive alternatives thanks to their reduced installation costs and invasivity [2].

A standard IoT system usually consists of a three-tiered architecture with: *i)* a bottom layer, in charge of sensing and actuation through devices that collect data for transmission to upper layers and perform actuation according to specified policies/commands (potentially sent back from upper levels of the stack); *ii)* a middle layer, encompassing different types of communication networks (e.g. local area networks, cellular networks or, more generally, the internet) and devices (e.g. routers, gateways) enabled by different communication technologies, ranging from Bluetooth to WiFi, from LTE to 5G; *iii)* a top application layer devoted to data storage, processing and analysis usually hosted on cloud computing platforms.

C. Machine Learning

Finally, once data is gathered from sensors (e.g. in the form of a time series) and is transmitted along the IoT communication pipeline, it is processed by means of machine learning techniques to perform inference and provide information (e.g. in the form of classification or regression) to downstream decision making. Deep learning represents the most recent forefront of this trend.

It should be remarked here the existence of an inherent tension between two different design strategies: on one hand, all data collected by a sensor can be transmitted remotely for its processing, on the other hand, it can be directly processed on board of sensor nodes. In the former case, data is sent for instance to edge devices, such as gateways, or to more powerful cloud computing hardware to which inference is delegated. In the latter case, machine learning models are used to perform predictions locally on peripheral devices. The above-mentioned tension is clearly the result of a trade-off among several variables, namely the energy spent for communication, the latency introduced by the processing/communication chain, and the accuracy achieved for solving a specific recognition task.

In this work, we introduce a vibration monitoring system purposed for SHM applications. The main goal of the proposed approach is to overcome the limitations of alternative solutions in terms of cost and flexibility (intended as ease of installation, maintenance, and operativity) through the design of a comprehensive system, which encompasses the following contributions:

- we design a novel Hall-effect based sensor to detect and measure vibrations; the resulting sensor is characterized by reduced costs and works at low frequency ranges (for instance it correctly detects peaks in the power spectrum of an earthquake waveform around 2.5Hz); the sensor is interfaced with off-the-shelf embedded platforms (i.e. a Raspberry PI) also characterized by reduced costs;
- we integrate the sensor node into an IoT framework that allows communicating readings to a cloud back-end system for visualization and processing purposes;
- we demonstrate the feasibility of performing complex machine learning inference directly on board of the adopted IoT hardware platform by implementing two different deep learning models (namely a convolutional and a recurrent neural network); non-trivial analysis of vibration signals by means of sophisticated deep learning models can be performed directly on board or offloaded to the cloud thanks to fully supported IoT functionalities;
- we provide an experimental assessment of the system under real-world working conditions in two use cases, namely the detection of a seismic event and an application for counting steps made by persons walking inside a public building; experimental results provide evidence of the versatility and flexibility of the proposed solution and its suitability as an alternative to other vibration monitoring systems.

The remainder of the article is organized as follows: in Section II we summarize the main state-of-the-art works in current scientific literature and frame their contributions with respect to our approach; in Section III we illustrate the proposed Hall-effect vibration sensor; in Section IV we describe the type and architecture of the two deep learning models used for performing inference; in Section V we introduce the setup adopted for the experimental validation of the proposed system and the related results obtained; in Section VI we conclude by recapitulating the main contributions and findings.

II. Related Work

The scientific literature about vibration monitoring in civil structures and infrastructures is particularly rich, being it a widely investigated subject of research.

A central assumption of SHM techniques is the possibility of monitoring the health status of a civil structure/infrastructure using vibration-based analysis. As a matter of fact, most SHM systems rely on the analysis of the so called-modal properites or vibration characteristics (i.e. natural frequencies, damping ratios, and mode shapes) which, in turn, depend on the physical features of the structure (i.e. mass, stiffness, and damping).

Since damaged structures change the physical properties of the structure, they result in modifications of the modal properties that can be used therefore as a proxy for damage detection [3]. Recent approaches have also introduced methods for detection and, at the same time, for the corresponding localization of possible damage within a structure, starting from the vibration data [4]. The main focus of our work is not the proposal of new methods for modal property analysis. Rather, we put forward a system-level design that could accommodate specific algorithmic approaches targeting vibration data analysis.

Importantly, many SHM methodologies nowadays make use of vibration data that can be extracted from MEMS sensors. These types of sensors, thanks to their increased precision, lower costs, and small size have progressively become a feasible alternative to traditional piezoelectric sensors [5], [2], [6]. In this article we describe a different approach for vibration sensing, based on a low-cost Hall-effect device which can be easily integrated into a wider IoT system.

Machine learning and, in particular, the deep learning paradigm, have increasingly gained popularity among scholars for vibration data processing [7]. For instance, in [8], the authors proposed a data-driven SHM method based on convolutional neural networks and fast Fourier transform to identify structural damage conditions from vibration data. An autoencoder architecture targeting nonlinear dimensionality reduction of input vibration signals has been recently introduced for the task of load identification in [9].

Another related line of research deals with the possibility of moving the execution of inference tasks (and also of learning tasks, in some cases) toward peripheral devices. Instead of relying on schemes entailing the transmission of whole data gathered from sensors to more powerful computing platforms (up to cloud systems), many works are currently investigating the opportunity to carry out specific machine learning tasks directly on board of sensor nodes or devices (such as gateways) located nearby in the network, according to *edge/fog computing* [10] or *tiny machine learning* paradigms [11], [12]. This potentially leads to benefits for security, privacy, and latency. The complex interplay between computation and communication also impacts energy consumption, which is a crucial factor to be taken into account in many IoT settings characterized by battery-operated devices[13].

Our contribution with respect to [8], [9] is represented by the implementation of two deep learning models on the embedded platform chosen as reference, hence showing the possibility of executing non-trivial machine learning tasks directly on the hardware platform in charge of collecting vibration data; given the integration with fully-fledged IoT protocols, the system can be reconfigured on the fly to also support cloud-enabled machine learning. We also benchmark the performances of the two implemented neural networks in a hypothetical smart building application (namely counting people's steps during walking).

Regarding the design of IoT systems for vibration monitoring, recently, Komarizadehal *et al.* have introduced a system based on an Arduino microcontroller equipped with five low-cost accelerometers [1]. The experimental assessment conducted by the authors with a comparison against two traditional piezoelectric sensors (as measured in terms of accuracy, resolution, and error) highlighted that the proposed platform can rival competitors, performing even better at low frequencies and low amplitude accelerations at a reduced (14x) cost [1]. In [14], an IoT sensing system for monitoring vibrations induced by construction sites is presented. The system, based on a Raspberry PI embedded platform and MEMS accelerometer is connected to the cloud via a USB dongle for 4G connectivity. Cloud-based frontend and backend complete the architecture implementing data storage and visualization (via MySQL database and web interface) and alarm detection based on predefined thresholds on the vibration signals.

Differently from our proposal, these works ([1], [14]) do not take into account any machine learning technique, while we also explore the implementation of machine learning models for vibration signal processing applications.

To conclude, the multifaceted field of SHM presents various challenges to be tackled by research toward the design of effective monitoring and diagnostics systems. Our proposal attempts to bridge some gaps in the literature by contributing a low-cost solution based on a magnetic sensor device interfaced with a widely used commercially available hardware; the proposed solution supports machine learning inference on the device, as we demonstrate by embedding two deep learning models on it, while it also allows full integration with state-of-the-art IoT communication protocols to support cloud computing.

III. The Proposed Hall-Effect Vibration Sensor

A Hall-effect sensor is a type of sensor in which the output voltage is directly proportional to the strength and direction of the magnetic field in which it is immersed. Its working principle is based on the Lorenz force inducted by the presence of a magnetic field perpendicular to the direction of an electrical current applied to a thin strip of metal. As the result of the Lorenz force, a difference in electric potential (voltage) between the two sides of the strip can be measured which is proportional to the strength of the magnetic field [15]. As a special feature, a hall-effect sensor responds also to a static (non-changing) magnetic field differently from inductive sensors, which respond only to changes.

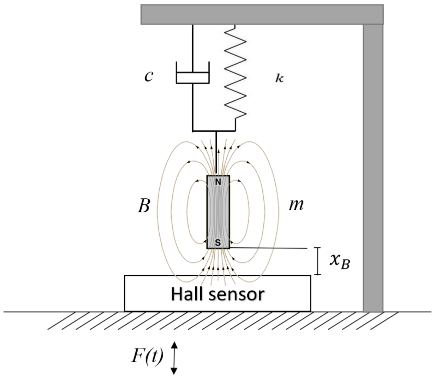


Fig. 1. The schematic representation of the proposed building vibration system consisting of a mass m, a spring k, a damping element c, and a magnetic field B positioned at a distance x_B from the hall-effect sensor

Hall sensors are traditionally used for proximity sensing, positioning, speed detection, and current sensing applications, or, combined with threshold detection, they act as a binary switch. In automotive, they are commonly used to time the speed of wheels and shafts, such as for internal combustion engine ignition timing, tachometers, and anti-lock braking systems.

In this work, we present a new low-cost IoT system, based on the Hall-effect, capable to measure building vibrations. In particular, we exploit the ability to produce an output proportional to the magnetic field of the Hall-sensor to dynamically monitor the displacement of an inertial static magnet with respect to the sensor which is rigidly anchored to the building.

Fig. 1 shows the main functioning principle of the proposed sensor. Notice that, for sake of simplicity the figure shows only the vertical axis but the same reasoning can be extended to other axes. The system can be led back to a classical damped harmonic oscillator consisting of a mass m suspended on a spring characterized by a constant k together with a damping element with a viscous damping coefficient c. Moreover, in the proposed system, the suspended mass consists of a static magnet with a magnetic field B positioned at a distance x_B from a Hall-effect sensor. When the building is subject to vertical vibrations an external force $F(t)$ is applied to the system which can be described, according to Newton's second law, by:

$$F(t) - kx - c\frac{dx}{dt} = m\frac{d^2x}{dt^2} \tag{1}$$

which can be rewritten into the form

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = \frac{F(t)}{m} \tag{2}$$

where $\omega_0 = \sqrt{\frac{k}{m}}$ is called the undamped angular frequency of the oscillator and $\zeta = \frac{c}{2\sqrt{mk}}$ is the damping ratio.

The value of the damping ratio ζ critically determines the behavior of the system. A measuring system like the proposed one, must fall into the case called "underdamped" with $\zeta < 1$. Under this condition The system oscillates with the amplitude gradually decreasing to zero [16]. The angular frequency of the underdamped harmonic oscillator is given by

$$\omega_1 = \omega_0\sqrt{1-\zeta^2} \tag{3}$$

while the exponential decay of the harmonic oscillator is defined by

$$\lambda = \omega_0\zeta \tag{4}$$

Considering a sinusoidal driving force defined as

$$F(t) = F_0 \sin(\omega t) \tag{5}$$

where F_0 is the driving amplitude, and ω is the driving frequency of the sinusoidal driving mechanism, equation 2 can be rewritten as:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = \frac{1}{m}F_0 \sin\omega t \tag{6}$$

The general solution at the steady state that is independent of initial conditions and depends only on the driving amplitude F_0, driving frequency ω, undamped angular frequency ω_0, and the damping ratio ζ. In particular, the steady-state solution is proportional to the driving force with an induced phase change φ:

$$x(t) = \frac{F_0}{mZ_m\omega}\sin\omega t + \varphi \tag{7}$$

where the value of the impedance of the system Z_m is defined as:

$$Z_m = \sqrt{(2\omega_0\zeta)^2 + \frac{1}{\omega^2}(\omega_0^2 - \omega^2)^2} \tag{8}$$

and the phase of the oscillation of the driving force φ is given by:

$$\varphi = \arctan\left(\frac{2\omega\omega_0\zeta}{\omega^2 - \omega_0^2}\right) + n\pi \tag{9}$$

Moreover, since the amplitude of the driving force $F_0 = m$ü where ü is the acceleration of the ground, the equation 7 can be rewritten as:

$$x(t) = \frac{\text{ü}}{Z_m\omega}\sin\omega t + \varphi \tag{10}$$

From 10 results that the value of $x(t)$ is proportional to the ground acceleration so measuring it entails measuring the ground acceleration. Furthermore, in the proposed system, $x(t)$ describes x_B which is the distance between the hall sensor and the magnet and, since the measured voltage on the Hall sensor (V_{Hall}) is proportional to the magnetic field B, which, in turn varies with the square of the distance, we can argue that measuring the voltage produced by the Hall sensor allows to measure the ground acceleration.

A. The Hardware Prototype

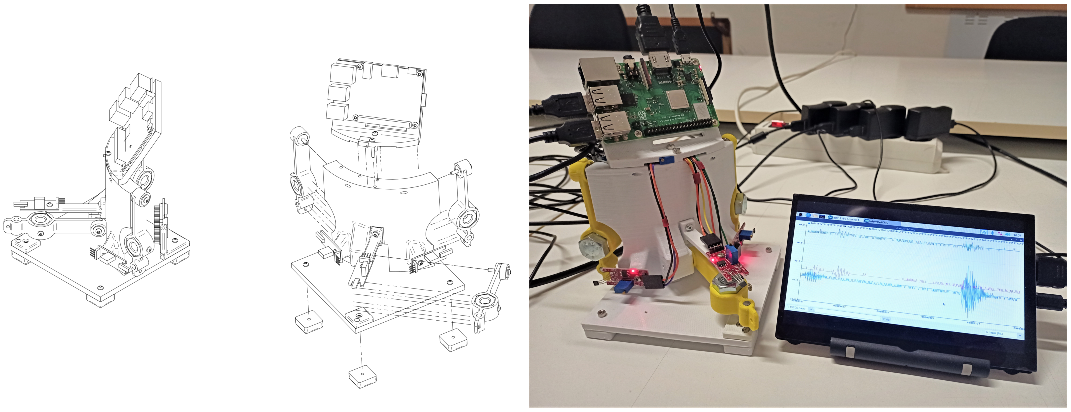
A prototype of the proposed system has been implemented by 3D printing a device capable of measuring building vibrations among the three orthogonal axes x, y, and z.

Fig. 2 shows the 3D solid model together with a picture of its realization. In particular, the central frame of the device sustains three orthogonal arms (yellow elements in the picture) that mount, at their ends, neodymium magnets. Each arm can only move along one degree of freedom as it is connected to the frame using a cylindrical bearing. The damping function is given by the bearing friction while the spring component has been implemented using magnetic repulsion with other neodymium magnets. Notice that, a calibrated mass has also been added to each arm since the very low mass of the magnets does not satisfy the design specifications. Finally, an analog linear Hall-effect sensor (Joy-it KY-024) has been placed perpendicularly and close to each suspended magnet to react at each displacement. The Hall-effect sensors are continuously sampled by means of a multichannel 16bit ADC (Az-Delivery ADS1115) which sends data to a Raspberry Pi model 3B+ through the I2C port.

Fig. 3 shows the circuit diagram together with a virtual representation of the electronic components. Since Hall-effect sensors produce an output also at a steady state (i.e. without any vibrations) the ADC has been connected in differential configuration so that it measures the difference between the sensor output and a reference value produced by means of the potentiometer R1. Tuning the potentiometer at a steady state allows calibrating the system to match the ADC specifications and to avoid ADC saturation.

B. The Management Software

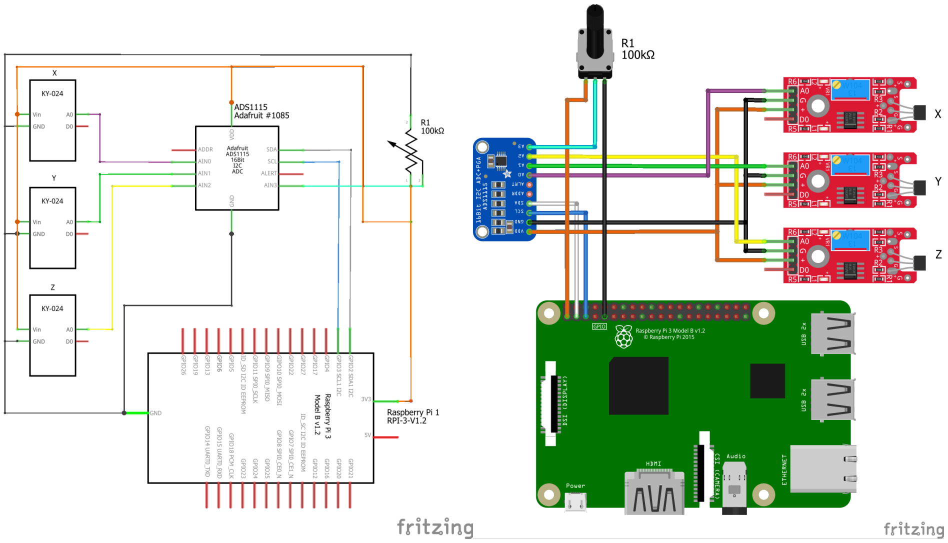
The proposed hardware is managed by a python process installed as a daemon service in the Raspberry Pi which collects, elaborates, and sends data to a time series database.



(a) 3D solid model

(b) Picture

Fig. 2. A 3D solid model of the proposed system (a) together with a picture of its implementation (b)



(a) Circuit diagram

(b) Circuit virtual representation

Fig. 3. A circuit diagram (a) and virtual representation (b) of the proposed sensor highlighting the single-board computer (raspberry Pi 3), the multichannel 16-bit ADC (ADS1115), and the three Hall-effect sensors (KY-024)

In particular, a first thread continuously samples the three ADC channels at a frequency of 50 Hz while a second thread elaborates and sends it to the remote server implementing a producer-consumer paradigm. In the current implementation, the consumer thread calculates the modulus of the three components of the signal and then computes the Root Mean Square (RMS) over a time window of 10 seconds. The RMS value is sent to the remote server while the entire buffer (10 seconds) containing the three signal components is stored only if the RMS exceeds a predefined threshold to avoid saving not significant data.

An autonomous monitoring system should guarantee the continuous collection of data avoiding as many as possible downtimes during which significant data could be lost. In a complex embedded system like the proposed one, several conditions can lead to the interruption of operation and the consequent loss of data such as, for example, a reboot of the Raspberry Pi due to a glitch or a loss of the power supply, a crash on the python process, a problem on the I2C communication with the ADC module or a network down. To guarantee the best reliability of the monitor activity, the management software exploits several recovery strategies. First, installing the management process as a daemon service guarantees that the Linux OS automatically starts it at boot time or restarts the process in case of termination. Moreover, to recover the monitoring activity in case of a lack of data from the ADC, we activate a watchdog timer that reboots the platform if no data arrives for more than 10 seconds. Finally, the consumer thread has been designed to save data locally during network problems and, eventually, reestablish server communication when the network will work again.

IV. Adding Machine Learning Models

To evaluate the suitability of the proposed system for the integration of machine learning algorithms, we tested two different supervised deep learning models, namely a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network aimed at recognizing a well-defined pattern, such as the vibrations induced by a walking person, within the time series collected by the system.

A. Deep-Learning Background

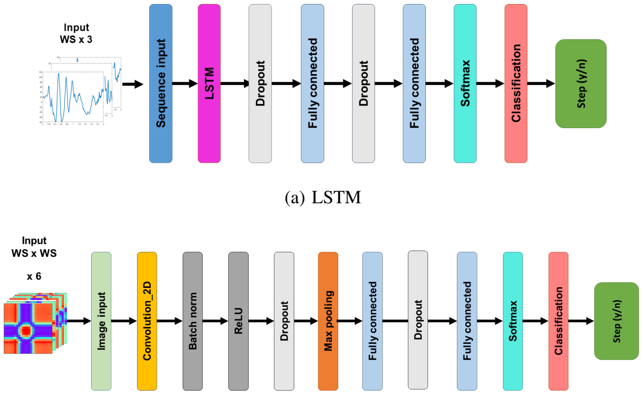
A CNN is typically composed of: *i)* a convolutional layer, *ii)* a pooling layer, *iii)* an activation function, *iv)* a fully connected layer, and *v)* a classification layer and it is traditionally used with very satisfying results in the field of image recognition and classification [17]. The convolutional layer takes as input a multi-channeled image, in n-dimensional format, and outputs a feature map. The pooling layer is in charge of keeping meaningful features while decreasing the size of the feature map. The activation layer allows us to learn other things, and its main role is to map the input to the output; the most well-known activation functions are: *Sigmoid*, *Tanh*, and *ReLU*, to cite a few. The fully connected layer acts as a classifier, and is normally placed at the end of the network, typically after the (last) pooling layer; the peculiarity of the fully connected layer is that each neuron is connected to all neurons of the predecessor layer. The final classification is however carried out by the classification layer, which is also responsible for error evaluation during training, i.e., for computing the difference between the predicted output and the actual one via the loss function. The most commonly adopted function is *Softmax*, which provides a prediction probability distribution.

It is worth noting that normally two-dimensional (2D) convolutional filters are used by CNN models to process 2D images; the opportunity to convert time series signal data to images is gaining a lot of momentum because it allows the application of computer vision techniques and performs classification tasks. In time series classification and pattern-recognition, this means that signals gathered from a triaxial sensor can be properly re-coded to images so that a “visual” analysis can be carried out to recognize, learn and classify patterns. Several re-coding techniques exist [18], [19], [20], and in this paper we leverage the deep learning approach proposed by [18], in which time series are converted to Gramian Angular Summation/Difference Fields (GASF/GADF) images. Such an approach envisages the representation of time series as a polar coordinate system that produce one image for the GASF and the other for the GADF. The main advantage of such an approach is that temporal and spatial relations are preserved.

Another deep learning approach used in time series classification is the LSTM model, which represents a recurrent neural network system. As detailed in [21], several versions of LSTM exist, but the most popular version is known as vanilla LSTM. A single LSTM unit is composed of a cell and three gates: input, output and forget gate. The cell is responsible for keeping track of values over time; the input gate is responsible for combining the current input, the output of the previous LSTM unit, and the value of the cell in the previous iteration in order to decide whether to select the potential candidate values to be added. The forget gate considers current input, state on memory cells and output at previous time step in order to decide which information should be removed from previous cell states. The output gate computes the output (to be sent to the output block) by combining the output of that LSTM unit at previous time, the current input and cell value in the previous iteration. LSTM networks are able to capture temporal information from time series data; since CNN networks are able to automatically extract significant features, combining both networks to build a hybrid architecture can bring benefit in terms of performance and accuracy in time series pattern-recognition.

B. The Proposed Deep-Learning Models

Fig. 4 shows the architecture of the proposed LSTM (a) and CNN (b) models.



(a) LSTM

(b) CNN

Fig. 4. The proposed deep learning models architecture i.e. (a) the LSTM network and (b) the CNN

Regarding the LSTM network, it takes in input the three signals gathered from the triaxial hall-sensor appropriately divided into windows of size equal to WS. These signals are arranged by a sequence input layer in a three-channel data chunk that is forwarded to an LSTM layer. The output of this layer is sent to two layers of fully connected neurons that act as a multi-layer Perceptron. Notice that, the two neurons layers are interspersed with two dropout layers, with a dropout probability 0.2, which helps prevent overfitting. The proposed network then terminates with a standard layer that computes the softmax function used by the classification layer to calculate the cross-entropy loss and to infer the presence or not of steps in the signal.

The proposed CNN model makes use of a the two-dimensional convolutional layer that acts on the six images obtained after encoding the three-time series into square images as described in Section IV-A. The images are then managed by an image input layer before going through the network structure. The convolutional layer is followed by a series of layers consisting of normalization, ReLU, dropout, and max-pooling layers. As in the previous network, the last part of the model is built by two neuron layers, a softmax, and a classification layer.

V. Experimental Setup and Results

In this section, we describe the experimental setup and we show the results obtained during two different sets of experiments, namely the building vibration monitoring and the machine learning-driven recognition of people's steps using the vibrations induced on the building floor.

A. Experimental Setup

The Raspberry Pi model 3B+ was running the original Raspberry Pi OS (Raspbian) with an updated version of the Python3 interpreter. To manage the ADS115 16bit ADC we installed the $adafruit - circuitpython - ads1x15$ library configured at the maximum data rate (860 samples per second). Thanks to the differential reading and the offset tuning circuit, the ADC gain was set to the maximum value which allows measuring in the $+/-0.256V$ range corresponding to a resolution of $0.002V/bit$.

In order to store, visualize, and process the collected data we built a time-series database by installing the InfluxDB open-source software on a desktop machine equipped with an Intel® Core i5 processor and 16GB of main memory running Ubuntu 22.04 LTS [22].

Concerning the machine learning experiments, the network models have been implemented, trained, and tested on the Matlab2022a® platform running on a Windows® desktop pc equipped with an Intel® Core i9 and 16GB of RAM. Moreover, once the models have been trained we converted them to C-language using the Matlab code-generation routine. The code was then compiled and executed on the Raspberry Pi to test its effectiveness and to measure the inference time. To provide a labeled dataset to the supervised classifier we manually annotated about 20 hours of collected traces with a binary label reporting the presence or not of vibrations induced by people's steps. Then the classification performances of the proposed classifiers have been measured using the following quantities:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F_1 score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

where TP are the true positives, TN the true negatives, FP the false positives, and FN the false negatives.

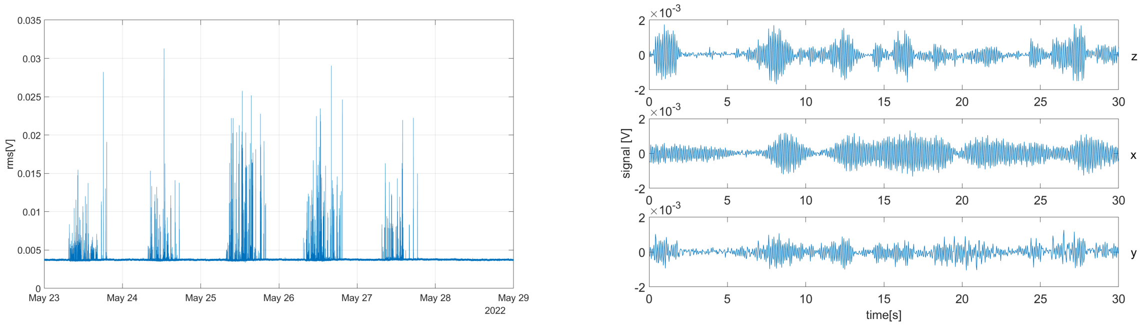
B. Measuring Building Vibrations

The prototype of the proposed monitoring system has been installed on a building named Collegio Raffaello located in the center of Urbino city - Italy. This building covers an area of about 2,600 square meters and it was built at the beginning of the eighteenth century by the will of Pope Clement XI. The building floors are mainly made of wooden beams covered by ceramic tiles. Nowadays, the ground floor of the Collegio Raffaello holds some craft stores and a bar while on the first floor takes place the Physics Museum of the University of Urbino. Lastly, the second floor, where the proposed system has been positioned, currently houses the degree program in Applied Computer Science and the degree program in Foreign languages and cultures of the University of Urbino so that, during the lesson period, it is frequented by many students and teachers.

Fig. 5(a) shows the data extracted from the InfluxDB database which represents the RMS calculated over a 10 seconds time window during a week on May 2022. Traditionally the University lessons are held only from Monday to Tuesday and are organized in units of two hours where the first one starts at 9.00 a.m. and ends at 11.00 a.m.. Then a second unit begins which ends with the lunch break at 1.00 p.m. Finally, in the afternoon, there may be one or more units starting at 2.00 p.m. The lesson activity exactly matches the peaks reported in the figure which shows high RMS values during the daytime hours of the five days of activity while, in the last two days of the week, the building vibrations are practically absent. Furthermore, the vibrations recorded are maximum in central days (Wednesday and Thursday) as happens for the occupation of the building.

Since the system also sends to the remote server the single waveforms relating to the x, y, and z axes during vibrational events that exceed the set threshold, it is possible to extract these traces for subsequent processing. Fig. 5(b) shows the raw signals from the Hall-effect sensors expressed in volts which have been collected when a single person was walking on the floor near the system. As expected, the greatest activity is recorded on the vertical axis (z) due to the transfer of the weight of the person on the floor, but it is nevertheless interesting to note that even in the x and y directions, which are parallel to the plane, it is possible to record non-negligible vibrations due to the walking activity.

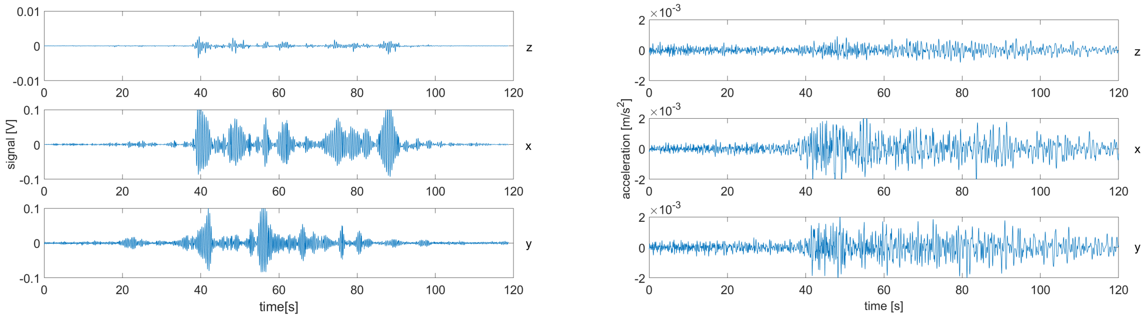
During continuous monitoring of the prototype a strong earthquake (Mw 5.5) struck Bosnia and Herzegovina region, causing the death of one person, the injury of at least two others, and forcing hundreds of people fleeing from their homes. Despite the distance that separates Urbino from the epicenter (about 450Km), the proposed monitoring system was able to record the waveforms related to the event for a duration of over 100 seconds. The raw signals are reported in Fig. 6 together with the ground acceleration collected by the official seismic station of the Italian National Seismic Network located at Monte Paganuccio (the closest station to Urbino). Despite the visible differences, it is interesting to note how the trend of the waveforms has several common points. For example, in both cases, the magnitude of the signal appears to be greater in the x and y axes, which are parallel to the floor, rather than in the vertical one. Moreover, both systems measure a vibration activity that lasts about 60 seconds. Clearly, a punctual comparison between the two systems, in addition to being out of the scope of this paper, is not possible for the fact that the system we proposed was located on the second floor of a building while the seismic station is directly supported on the ground so that the vibrations recorded by our system are, in effect, those induced on the building by the motion of the



(a) Building vibrations during a week

(b) Signals components during walking

Fig. 5. Measured RMS of the building vibrations during an entire week (a) and triaxial components of the vibrations collected during single person walking



(a) Proposed system

(b) IV:MPAG seismic station

Fig. 6. Vibration waveforms measured during bosnia earthquake of April 22, 2022 (Mw 5.5) by the proposed system (a) and by the the IV:MPAG seismic station (b)

earthquake. Despite this, the comparison allows us to argue that the proposed system is sufficiently sensitive to low-entity vibrations and therefore capable of monitoring the vibrations to which a building is subject.

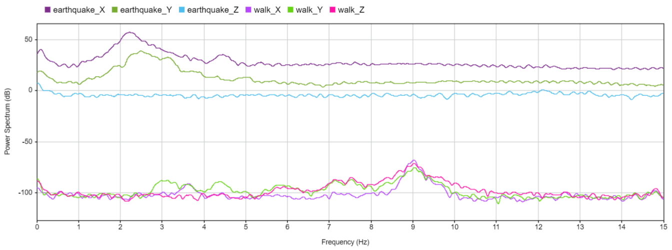


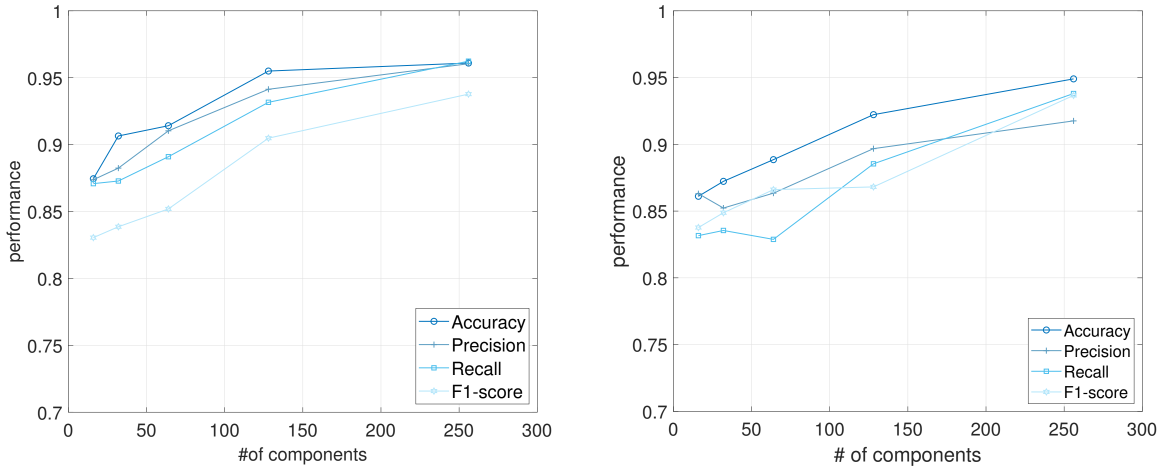
Fig. 7. Comparison of the spectral analysis of waveforms collected during an earthquake and single person walking

Finally, to evaluate the system response to the different patterns and kinds of vibrations, we report in Fig. 7 the power spectrum of the earthquake waveforms together with the spectrum of the vibration induced by a walking person. The first evident thing is that the two signals have very different predominant frequencies, for instance, the earthquake shows a main peak between 2 and 3 Hz (compatible with that detected by the seismic station) while the walking vibrations around 9 Hz.

C. Detecting People's Steps

Starting from the labeled dataset, the two proposed models have been tested when varying the number of different components. The training set and test set have been split according to a holdout cross-validation methodology with 75% of the examples used for training and 25% kept for testing. Notice that, for a convolutional layer, this number represents the convolutional filters, for the LSTM layer the number of hidden units (i.e. the amount of information remembered between time steps), and for the fully connected layer the number of hidden neurons. Fig. 8 reports the performance measured after training and testing each configuration of the proposed models obtained when varying the number of internal components. As expected, both for LSTM and CNN, increasing the number of components increases the classification performances reaching a very high value (up to 96% of accuracy for LSTM). Each model was then ported on the Raspberry Pi to characterize it in terms of memory footprint and inference latency.

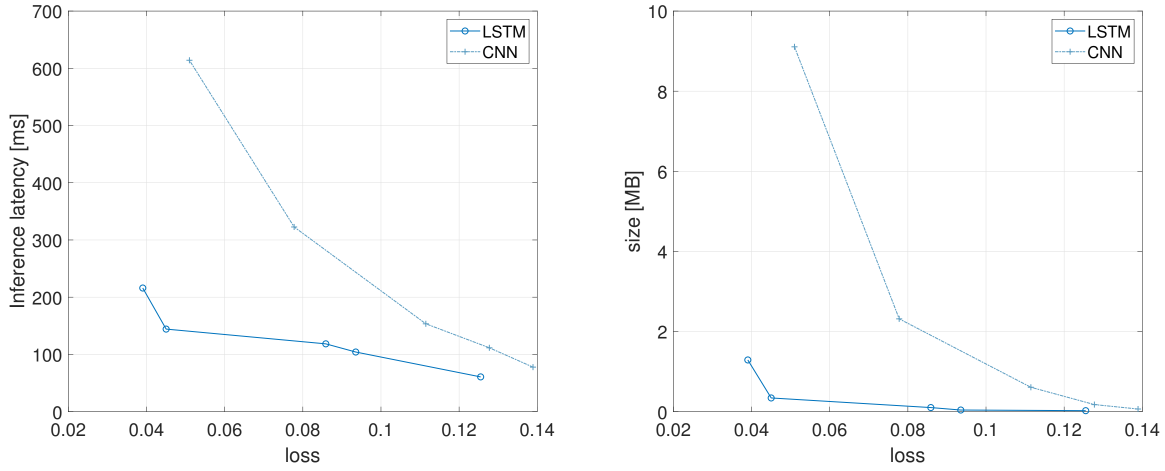
Table I reports the complete models characterization when varying the number of components. The increasing complexity of the models negatively reflects on the size and timing. Interestingly, the LSTM network outperforms the CNN both in terms of classification accuracy and memory and time requirements. For instance, the training time of CNN is 5× greater, the inference latency reaches about 3×, and the memory footprint is 7× greater with respect to the LSTM. Notice



(a) LSTM

(b) CNN

Fig. 8. Performance metrics of the LSTM (a) and CNN (b) networks when varying the network dimension



(a) Loss Vs latency

(b) Loss Vs size

Fig. 9. Pareto curves of classification loss Vs inference latency (a) and model size (b)

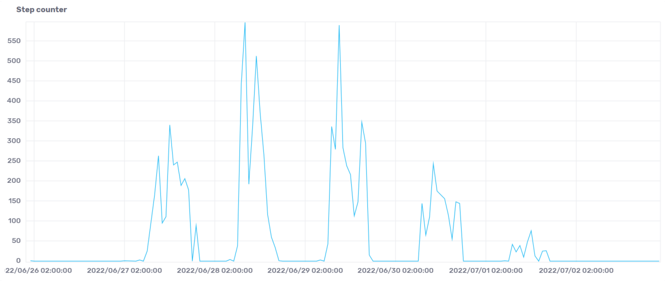


Fig. 10. The number of steps measured during a week on the collegio raffaello building

that, for real-time requirements the last CNN configuration (i.e. # of components = 256) cannot be used on Raspberry Pi as the inference latency would exceed the size of the monitoring window (500ms) thus making the consumer thread systematically slower than the producer with consequent data loss.

In Fig. 9, the Pareto curves obtained by plotting the classification loss Vs the inference latency (a) and model size (b) are reported. It is evident that the high classification performance combined with the low inference latency and the very low memory occupation make the LSTM network the best compromise for the detection of people's steps starting from the vibrations induced on the building using the proposed system.

According to these results, we permanently installed the trained LSTM on our system prototype with the scope of measuring the walking activity on the building floor.

TABLE I. CNN AND LSTM ACCURACY, TRAINING TIME, LATENCY, AND MEMORY FOOTPRINT AS FUNCTION OF NUMBER OF COMPONENTS

# of components	Accuracy		Training T [s]		Latency T [ms]		Size [kb]	
	LSTM	CNN	LSTM	CNN	LSTM	CNN	LSTM	CNN
16	0.874	0.861	950	4534	60.7	77.6	25	68
32	0.906	0.872	818	4398	103.2	111.5	42	184
64	0.914	0.889	844	4449	114.8	153.4	106	638
128	0.955	0.922	868	4623	144.3	322.4	358	2429
256	0.961	0.949	921	5127	217.2	614.6	1353	9550

Fig. 10 shows the number of steps measured during the last week of June 2022. Notice that, this week there was no teaching activity so the building was frequented only by teachers. In particular, the plot has been obtained using the influxdb server querying language to sum the number of counted steps over a time window of one hour. The maximum value of about 550 steps/hour was found on Tuesday and Thursday which are the days of greatest attendance by teachers while the total amount of recorded steps was about 6,500 steps.

VI. CONCLUSIONS

Vibration monitoring systems are traditionally used to assess the vibration impact of natural and human activities on buildings, to evaluate the health condition of structures. However, classical vibration monitoring systems are associated with limitations such as expensive devices, difficult installation, complex operation, etc. In this paper, we presented a low-cost Internet of Things monitoring system based on Hall-effect sensors encompassing integration with deep-learning machine models. The results obtained in real-life experiments (namely, the detection of seismic events and the count of steps made by people during walking in a public building) demonstrate that the proposed system can effectively gather vibration data in a built environment and operate in a standard fashion by transmitting all data remotely toward cloud-base backend or, conversely, perform intelligent pattern recognition tasks directly on the end device with accurate performance levels and reduced latency. We may therefore conclude that the proposed solution could represent a versatile and promising alternative to traditional vibration monitoring systems.

REFERENCES

[1] S. Komarizadehasl, B. Mobaraki, H. Ma, J.-A. Lozano-Galant, and J. Turmo, “Development of a low-cost system for the accurate measurement of structural vibrations,” *Sensors*, vol. 21, no. 18, p. 6191, 2021.

[2] F. Di Nuzzo, D. Brunelli, T. Polonelli, and L. Benini, “Structural health monitoring system with narrowband iot and mems sensors,” *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16 371–16 380, 2021.

[3] R. Astroza, H. Ebrahimian, J. P. Conte, J. I. Restrepo, and T. C. Hutchinson, “Statistical analysis of the modal properties of a seismically-damaged five-story rc building identified using ambient vibration data,” *Journal of Building Engineering*, vol. 52, p. 104411, 2022.

[4] Y. Liao, A. S. Kiremidjian, R. Rajagopal, and C.-H. Loh, “Structural damage detection and localization with unknown postdamage feature distribution using sequential change-point detection method,” *Journal of Aerospace Engineering*, vol. 32, no. 2, p. 04018149, 2019.

[5] S. M. Khan, M. U. Hanif, A. Khan, M. U. Hassan, A. Javanmardi, and A. Ahmad, “Damage assessment of reinforced concrete beams using cost-effective mems accelerometers,” *Structures*, vol. 41, pp. 602–618, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352012422003629

[6] F. Zonzini, M. M. Malatesta, D. Bogomolov, N. Testoni, A. Marzani, and L. De Marchi, “Vibration-based shm with upscalable and low-cost sensor networks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7990–7998, 2020.

[7] M. Flah, I. Nunez, W. Ben Chaabene, and M. L. Nehdi, “Machine learning algorithms in civil structural health monitoring: a systematic review,” *Archives of computational methods in engineering*, vol. 28, no. 4, pp. 2621–2643, 2021.

[8] Y. He, H. Chen, D. Liu, and L. Zhang, “A framework of structural damage detection for civil structures using fast fourier transform and deep convolutional neural networks,” *Applied Sciences*, vol. 11, no. 19, p. 9345, 2021.

[9] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, “An autoencoder-based deep learning approach for load identification in structural dynamics,” *Sensors*, vol. 21, no. 12, p. 4207, 2021.

[10] O. Gómez-Carmona, D. Casado-Mansilla, F. A. Kraemer, D. López-de Ipiña, and J. García-Zubia, “Exploring the computational cost of machine learning at the edge for human-centric internet of things,” *Future Generation Computer Systems*, vol. 112, pp. 670–683, 2020.

[11] 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.

[12] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang *et al.*, “Tensorflow lite micro: Embedded machine learning for tinyml systems,” *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, 2021.

[13] 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*, 2013, pp. 1–6.

[14] Q. Meng and S. Zhu, “Developing iot sensing system for construction-induced vibration monitoring and impact assessment,” *Sensors*, vol. 20, no. 21, p. 6120, 2020.

[15] E. Ramsden, *Hall-effect sensors: theory and application*. Elsevier, 2011.

[16] J. P. Den Hartog, *Mechanical vibrations*. Courier Corporation, 1985.

[17] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions,” *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[18] Z. Wang and T. Oates, “Imaging time-series to improve classification and imputation,” in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2015-January, 2015, pp. 3939–3945.

[19] G. Baldini, G. Steri, R. Giuliani, and C. Gentile, “Imaging time series for internet of things radio frequency fingerprinting,” in *2017 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2017, pp. 1–6.

[20] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K. R. Choo, “Imaging and fusing time series for wearable sensor-based human activity recognition,” *Information Fusion*, vol. 53, pp. 80–87, 2020.

[21] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[22] InfluxData. (2022) InfluxDB: Open Source Time Series Database. [Online]. Available: https://www.influxdata.com/