POLITECNICO DI TORINO
Repository ISTITUZIONALE

Foundation Models for Structural Health Monitoring

Original
Foundation Models for Structural Health Monitoring / Benfenati, L., Pagliari, D.J., Zanatta, L., Velez, Y.A.B., Acquaviva, A., Poncino, M., Macii, E., Benini, L., Burrello, A.. - In: IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. - ISSN 2377-3782. - ELETTRONICO. - 10:6(2025), pp. 1103-1115. [10.1109/tsusc.2025.3592097]

Availability:
This version is available at: 11583/3003612 since: 2025-10-03T16:22:15Z

Publisher:
Institute of Electrical and Electronics Engineers

Published
DOI:10.1109/tsusc.2025.3592097

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

Foundation Models for Structural Health Monitoring

Luca Benfenati, Daniele Jahier Pagliari, *Member, IEEE*, Luca Zanatta, Yhorman Alexander Bedoya Velez, Andrea Acquaviva, Massimo Poncino *Fellow, IEEE*, Enrico Macii *Fellow, IEEE*, Luca Benini *Fellow, IEEE*, Alessio Burrello, *Member, IEEE,*

***Abstract*—Structural Health Monitoring (SHM) is a critical task for ensuring the safety and reliability of civil infrastructures, typically realized on bridges and viaducts by means of vibration monitoring. In this paper, we propose for the first time the use of Transformer neural networks, with a Masked Auto-Encoder architecture, as *Foundation Models* for SHM. We demonstrate the ability of these models to learn generalizable representations from multiple large datasets through self-supervised pre-training, which, coupled with task-specific fine-tuning, allows them to outperform state-of-the-art traditional methods on diverse tasks, including Anomaly Detection (AD) and Traffic Load Estimation (TLE). We then extensively explore model size versus accuracy trade-offs and experiment with Knowledge Distillation (KD) to improve the performance of smaller Transformers, enabling their embedding directly into the SHM edge nodes.**

We showcase the effectiveness of our foundation models using data from three operational viaducts. For AD, we achieve a near-perfect 99.9% accuracy with a monitoring time span of just 15 windows. In contrast, a state-of-the-art method based on Principal Component Analysis (PCA) obtains its first good result (95.03% accuracy), only considering 120 windows. On two different TLE tasks, our models obtain state-of-the-art performance on multiple evaluation metrics (R^2 score, MAE% and MSE%). On the first benchmark, we achieve an R^2 score of 0.97 and 0.90 for light and heavy vehicle traffic, respectively, while the best previous approach (a Random Forest) stops at 0.91 and 0.84. On the second one, we achieve an R^2 score of 0.54 versus the 0.51 of the best competitor method, a Long-Short Term Memory network.

***Index Terms*—Structural Health Monitoring, Traffic Load Estimation, Foundation Models, Masked Autoencoders, Deep Learning**

I. Introduction

Structural Health Monitoring (SHM) is crucial to ensure longevity and safety of critical infrastructures such as bridges, highways, and tunnels. In fact, these structures are subject to deterioration due to factors such as ageing, environmental conditions, and traffic loads, posing potential safety hazards and substantial financial burdens for maintenance and repairs.

Manuscript received January XX, XXXX; revised January XX, XXXX. (*Corresponding Author: Luca Benfenati).*

L. Benfenati, D. Jahier Pagliari, Y. Bedoya Velez and M. Poncino are with the Department of Control and Computer Engineering, Politecnico di Torino, Turin, 10129, Italy (e-mail: luca.benfenati@polito.it, daniele.jahier@polito.it; s287296@studenti.polito.it; massimo.poncino@polito.it).

L. Zanatta, A. Acquaviva, L. Benini, A. Burrello are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: luca.zanatta3@unibo.it; andrea.acquaviva@unibo.it; luca.benini@unibo.it; alessio.burrello@unibo.it).

E. Macii and A. Burrello are with the Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, 10129, Italy (e-mail: E. enrico.macii@polito.it; alessio.burrello@polito.it,).

L. Benini is with the Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zürich, Switzerland (e-mail: lbenini@iis.ee.ethz.ch).

SHM systems address this challenge by proactively and continuously monitoring infrastructure in real time, enabling early detection of anomalies and potential damages. This facilitates timely interventions that ensure public safety while optimizing resource allocation through cost-effective strategies. When applied to critical infrastructures such as bridges and viaducts, SHM can also exploit Traffic Load Estimation (TLE) as a metric that captures the dynamic nature of traffic loads that the structures endure, to continuously assess structural health and predict maintenance needs.

A typical SHM system comprises a network of sensors responsible for measuring multiple parameters relevant to the current state of the structure as well as its surrounding environment. The most widely used SHM sensing devices include fiber optic sensors [1], [2], accelerometers [3], [4], cameras [5] and strain gauges [6]. Among these options, networks of Micro-Electro-Mechanical Systems (MEMS) for vibration monitoring stand out due to their cost-effectiveness and relatively simple deployment in diverse environments. Data gathered by these accelerometers are then used as the input of a Machine Learning (ML) model to address tasks such as Anomaly Detection (AD) or TLE. These devices have demonstrated high accuracy compared to other technologies on several SHM tasks [7].

ML approaches for vibration-based SHM span a diverse range of techniques, from classic methods as in [8], [9], to more recent deep learning solutions such as Convolutional Neural Networks (CNNs) [10] and Autoencoders [11], [12]. While Transformer neural networks have revolutionized the Computer Vision and Natural Language Processing (NLP) fields, in SHM they have only been applied to camera-based monitoring solutions [13], and not yet (to our knowledge) to vibration-based ones.

In this paper, we introduce two key contributions in the direction of a new family of models for acceleration-based SHM. First, we explore, for the first time, the application of Transformers to this task. Second, building on the recent breakthroughs of foundation models across diverse domains (language, vision, etc.) [14], we explore self-supervised learning to take advantage of large amounts of easily available unlabelled data and develop general backbone feature extractors, which can be then fine-tuned for multiple SHM tasks. By doing so, we outperform current, single-task ML approaches, which are instead trained on small datasets in a fully supervised way.

More specifically, we adopt a transformer-based Masked Autoencoder architecture inspired by [15], which has become a standard in self-supervised learning, demonstrating better performance compared to classical autoencoder architectures.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

This approach demonstrates the potential of foundation models for advancing data-driven SHM, particularly in overcoming the challenge of collecting large amounts of labeled data, a process that is typically time-consuming and costly, requiring extensive instrumentation of target infrastructures with additional sensors, such as cameras or Weigh-in-Motion (WiM) systems [16] to obtain ground truth data.

In summary, the contributions of this work are:

- We consider three different SHM datasets, including a newly collected one, larger than those introduced in previous works and fully labelled. These datasets are collected by a specialized company with patented sensor networks, expert in real-time SHM. They include two viaducts and a bridge, representative of common Italian infrastructures.
- We build a Transformer-based Masked Autoencoder, inspired by [15], as the first Foundation Model for SHM, capable of addressing multiple tasks within a unified framework. By pre-training on three diverse datasets without labels (self-supervised learning) and fine-tuning on each task, we outperform individually trained models, demonstrating the benefits of transferable representations. This two-stage training approach simulates the realistic SHM scenario where large unlabelled datasets are easily available, but labelled samples are scarce.
- Through our experiments, our fine-tuned models outperform state-of-the-art algorithms on all three datasets. In particular, on the first dataset, we achieve an AD accuracy of 99.92%, a sensitivity of 100%, and a specificity of 99.9%, with respect to state-of-the-art 75.76%, 55.68%, and 98.75%. On the other two datasets, and on three TLE variants, we achieve an R^2 score of 0.97, 0.85, and 0.54 respectively, outperforming the state-of-the-art that stops at 0.91, 0.84, and 0.10.
- We carry out an extensive search on the optimal model size. In this context, we test Knowledge Distillation (KD) to train smaller models to imitate larger ones, ultimately targeting deployment on resource-constrained nodes for real-time SHM at the edge. Results show that distilled models often outperform standardly fine-tuned and equally sized counterparts on downstream tasks.
- We demonstrate the feasibility of deploying our models on a gateway-class edge device, i.e., the NVIDIA Jetson Nano, showing a latency well below the real-time constraint of the three analyzed use cases.

The rest of the paper is organized as follows. Section II provides the necessary background. Section III overviews the SHM literature, describing the different approaches to AD and TLE. Section IV describes the three considered viaducts, the corresponding SHM sensor networks, and the data acquisition and labeling processes. Section V details our foundation model approach, presenting the processing pipeline, the architecture of the model, and its training procedure. Section VI presents the experimental results and Section VII concludes the paper. We open-source our code and pre-trained models at `https://github.com/eml-eda/tle-supervised`.

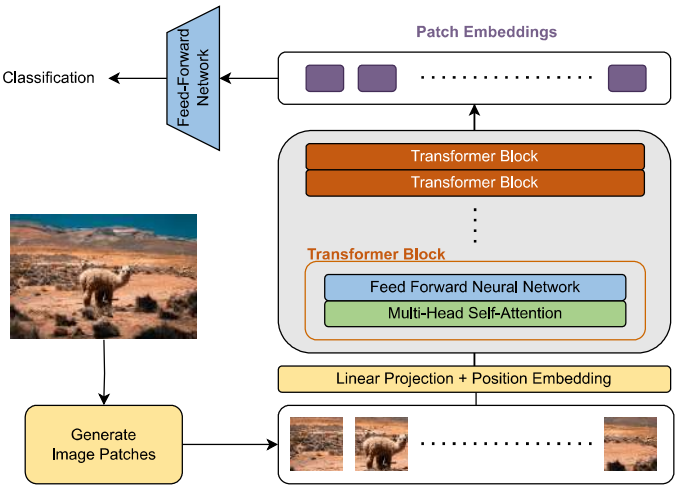


Fig. 1. Vision Transformer ViT Architecture

II. Background

A. Transformer Neural Networks

The Transformer architecture was introduced in 2017 [17], ushering in a new era for NLP applications and Deep Learning (DL) in general. The architecture leverages an encoder-decoder structure with stacked self-attention and feedforward layers, enabling the model to process input sequences and generate outputs (e.g., translations) autoregressively. Its core innovation lies in self-attention, which allows the model to directly attend to relevant parts of the input sequence, capturing long-range dependencies. Scaled dot-product and Multi-Head Self-Attention (MHSA) enable efficient and diverse attention computation, while positional encodings address the lack of inherent sequence awareness [18]. Its impact goes beyond NLP, making it a cornerstone of modern DL due to its scalability, parallelizability, and ability to model long-range dependencies. In our work, we use an architecture derived from the Vision Transformer (ViT) [19], which first applied an attention-based Transformer to image processing. This is achieved by breaking the image into fixed-size patches, linearly embedding each of them, adding the position embeddings and then feeding the resulting sequence of vectors to a standard Transformer encoder. A simplified scheme of a ViT encoder architecture is shown in Figure 1. Notably, we apply a similar architecture to vibration data rather than standard RGB images, after extracting their spectrogram, as detailed in Sec. V-A.

B. Masked Autoencoders

Masked autoencoders sparked renewed interest in 2021 when [20] successfully adopted them as scalable self-supervised learners for Computer Vision applications. They are encoder-decoder models, whose encoder is the ViT shown in the previous section. The decoder, instead, takes the patch embeddings as inputs and is tasked to reconstruct the original image. During training, some image patches are randomly masked before being fed to the network, forcing the autoencoder to reconstruct the missing information solely from the remaining visible parts. This process encourages the model

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

to learn robust and diverse representations of visual features. The simplicity and effectiveness of this scheme have led masked autoencoders to achieve state-of-the-art performance in various downstream tasks, demonstrating their potential as powerful tools for visual representation learning [18]. The specific masked autoencoder architecture used in our work will be detailed in Sec. V-B and shown in Fig. 4.

C. Foundation Models

The term Foundation Model refers to a DL model that is trained on broad data, using self-supervision at scale, and can be adapted to a wide range of downstream tasks [14]. The pre-training phase is usually carried out on diverse and unlabelled data, to produce general-purpose data representations that, unlike specialized models trained for specific tasks, serve as a foundational block. Fine-tuned on specific downstream tasks, foundation models offer several advantages: increased training efficiency, broader applicability, and the ability to transfer learned knowledge across domains. This shift in AI from task-specific models to a reusable "foundation" approach has opened new avenues for leveraging the power of DL across various applications, from tasks like summarization and translation in NLP [21] to image classification and object detection in Computer Vision [22], [23], and time-series forecasting [24], [25]. However, the adoption of this novel paradigm in other domains, including SHM, is still unexplored.

III. Related Works

This section presents the current state-of-the-art approaches to SHM, including both AD (Sec. III-A) and TLE (Sec. III-B). For each task, we first broadly overview works that use other sensors, and then focus on accelerometer-based methods.

A. Anomaly Detection

Concerning visual AD, [13] applies a ViT-like approach for semantic segmentation to identify damaged structural components in a railway viaduct, achieving 97% and 90% mIoU for component and damage segmentation, respectively, and outperforming existing vision-based methods. Similarly, [26] detects bridge damage using a roving multi-camera system, comparing five unsupervised techniques (Autoencoder, K-Nearest Neighbours, Kernel Density, Local Outlier Factor, and Isolation Forest) and achieving F1 scores between 0.96 and 0.97 in controlled tests. While visual approaches reach impressive accuracy, they incur high installation costs and are effective only under good visibility conditions (e.g., not during night or foggy days).

Therefore, most AD methods for SHM rely on acceleration data. For example, [27] employs a 1-dimensional CNN to estimate the Probability of Damage across nine scenarios of increasing severity. In [10], statistical features (maximum, minimum, mean, variance, skewness, kurtosis) extracted from acceleration data feed into a CNN with three convolutional layers, global average pooling, and softmax, achieving 97.6% accuracy on a long-span cable-stayed bridge dataset. Spiking neural networks have also been explored in [28], [29], reaching 95% accuracy and Matthews Correlation Coefficient (MCC) values of 0.88 for viaduct damage classification. These supervised approaches require extensive labeled data, limiting their scalability. Furthermore, they are specifically trained on a single infrastructure, while their generalization over multiple ones is not tested.

In contrast, unsupervised methods such as [30] use Autoencoders to extract features from vibration responses for indirect damage diagnosis, though predictions can deviate by an average of five levels on a 30-level scale. Similarly, [31] employs an unsupervised Recurrent Neural Network (RNN) for damage detection and localization, achieving 93% and 85% accuracy, respectively, albeit on simulated data. The most promising unsupervised AD method for vibration data is presented in [32], which leverages PCA compression without labeled anomalies. PCA coefficients are fitted solely on healthy data, so anomalous vibrations are poorly reconstructed and flagged when the reconstruction error exceeds a threshold. On one of our real-world datasets—collected from a viaduct in northern Italy—this approach achieved 98.8% accuracy, 100% specificity, and 97.33% sensitivity, and is used as the state-of-the-art benchmark in our comparisons (see Section VI).

B. Traffic Load Estimation

Kamkar et al. [5] employ smart cameras to detect, classify, and quantify vehicles on public bridges, achieving 92.1% accuracy in high-traffic scenarios. However, their performance falls to 74.8% under low-light conditions, highlighting the sensitivity of visual-based methods to environmental factors. In contrast, methods using magnetic sensors with adaptive thresholds and classification trees [33], [34] report accuracy ranging from 84.7% to 99.9%, but they often achieve their best performance under controlled conditions (e.g., single, slow-moving vehicles) that do not reflect real-world traffic. Alternative sensor technologies, such as fiber optic [1] and infrared sensors [2], can also attain high accuracy; however, they also face challenges related to environmental sensitivity and high installation costs (they have to be embedded in the asphalt, requiring road interventions), limiting their feasibility in less populated areas. [35] develops an agent for adjusting traffic signals based on TLE data, but this approach does not directly address vehicle classification or quantification challenges. On the other hand, acceleration-based approaches for TLE are less common. In [36], the superiority of MEMS accelerometers is demonstrated, showing robustness to sensor placement and structural damage. In [37], TLE is framed as an anomaly detection problem, using k-means clustering to differentiate between light and heavy vehicles based on deviations from a periodically updated baseline. Although this method avoids reliance on labeled data, its performance in high-traffic conditions is insufficient. The most recent work, [8], compares several classical machine learning techniques and a deep learning approach (Multi-Layer Perceptron) on MEMS accelerometer data, achieving state-of-the-art R^2 scores of 0.91 for light vehicles and 0.84 for heavy vehicles. We use this as the benchmark for comparison in Section VI.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097



Fig. 2. A) Aerial view of the viaduct under analysis. B) Sensor positioning on prestressing cables. C) Prestressing cables.

IV. Structural Health Monitoring Installations

This section describes the three SHM Use Cases (UCs) considered in our work. The first UC targets anomaly detection, identifying abrupt structural changes, while the other two focus on traffic load estimation with continuous targets. Whereas these three scenarios only regard vehicle communication infrastructures, they serve as a preliminary validation of self-supervised transformer-based models for large-scale SHM systems, which could be extended in the future to other civil structures whose structural health is measured by similar acceleration signals.

A. Use Case 1 (UC1): Anomaly Detection Task

This UC utilizes vibration data from a dual-carriageway state highway viaduct in northern Italy. In 2019, a structural strengthening intervention on one of its spans altered the vibration signals. Before the intervention, five SHM nodes were installed under the bridge to monitor its vibrations, providing data from both pre- and post-intervention periods.

We treat post-intervention vibrations as normal and pre-intervention vibrations as anomalous, simulating abrupt structural changes (e.g., due to an earthquake). The sensor nodes use LIS344ALH 3-axis accelerometers [38] (±2/±6g range) and an HTS221 temperature/humidity sensor [39]. Data collection and processing are handled by an STM32L476VGTx microcontroller (ARM Cortex-M4, 80 MHz, 96 KB SRAM, 1 MB Flash).

B. Traffic Load Estimation Task

1) Use Case 2 (UC2): This UC focuses on a viaduct in Italy, monitored by an SHM sensor network since 2017 (Fig. 2). The viaduct spans 580 m, supported by five concrete pillars, with individual spans ranging from 67 m to 112 m. Each span consists of a steel caisson housing 12 prestressing cables, with cross-section heights varying from 6.0 m at the bearings to 3.0 m at mid-span.

Given that the structural integrity of prestressed bridges depends on cable durability, a sensor system (Fig. 2) was installed to detect breakages. Since September 2017, a network of 90 sensors has been actively collecting and storing data in the cloud. Each sensor node features an STM32F405RG microcontroller (MCU) for data acquisition and optional analytics, alongside the same accelerometer, temperature, and humidity sensors used in Section IV-A.

2) Use Case 3 (UC3): This UC examines a reinforced concrete girder bridge with 18 spans, totaling 583 meters in length, with two lanes. The bridge follows an isostatic static scheme, with 20-meter spans, except for the first (10 m) and last (29.5 m) spans. Since 2022, an industrial SHM system has continuously recorded data.

The system includes 282 MEMS biaxial inclinometers and 142 MEMS triaxial accelerometers (±2 g full scale, 100 Hz sample rate), evenly distributed across spans. Accelerometers monitor the two external beams per span, with sensors positioned at the quarter, third, and midspan locations.

Additionally, a Weigh-in-Motion (WiM) system is installed 600 meters before the bridge, providing ground truth traffic data—including lane, vehicle length, weight, speed, and axle count. To align WiM data with SHM sensor readings, a time offset correction is applied based on vehicle speed and distance between the WiM system and each sensor.

V. Methodology

This section presents our novel approach to vibration-based SHM leveraging self-supervised learning and Transformer foundation models. Firstly, we outline the data processing toolchain and train/test splitting (Section V-A). Then, we delve into our model architecture (Section V-B) and training strategies and model exploration (Section V-C, Section V-D). The core objective of our work is to understand whether it is possible to pre-train a large Transformer model on substantial amounts of unlabeled data and then fine-tune it to realize different SHM tasks. We consider both AD and TLE as downstream tasks to validate this hypothesis, focusing on all three Use Cases described in Sec. IV. Specifically:

- AD is formulated as a "one-class" classification, discriminating between normal and anomalous structural states, similar to the approach of [32].
- TLE is framed as a regression problem, predicting a scalar traffic value. For UC2, we split the estimation of light and heavy vehicle traffic into two separate tasks.

The complete pipeline is shown in Figure 3.

A. Pre-processing Pipeline

This section outlines the pre-processing applied before inputting data into our masked autoencoder. The pre-processing settings are consistent for training and validation data across all analyzed UCs, except when expressly indicated.

The first pre-processing step is **windowing**, i.e., the division of the time-series data into time windows, which the algorithm can process. For UC1, we use 5s windows, as in the reference paper [32], with a stride of 2s to increase the number of training samples. For the other use cases, we enlarge the window to

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

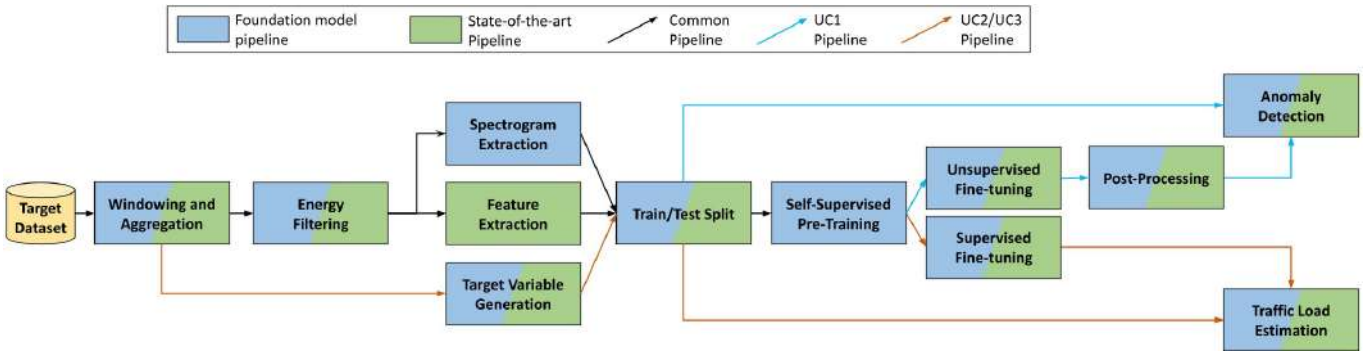


Fig. 3. Overview of the processing pipeline for traffic load estimation and anomaly detection. In blue are the blocks for our foundation model pipeline, and in green are the ones for SoA algorithms. Both colours are used for shared blocks.

60s since the reference study [8] reported the best performance with this dimension, using a 2s stride in UC2, and 15s stride in UC3, given the extremely lower traffic on the viaduct (with a 2s stride, the information contained in successive windows was almost identical, while causing a linear increase in training time). All windows are composed of uni-dimensional signals since we always employ either a single sensor or multiple sensors, but consider them separately. We also only consider the z-axis of the acceleration data. Thus, each window is always a single vector of dimension $T = 100 \times T_{win}$, where 100 is the acceleration sampling frequency in all three datasets. All time windows are normalized with mean and standard deviation.

After creating the windows, we apply **energy filtering**, i.e., we disregard as irrelevant those windows whose cumulative energy is under a certain threshold, $th = 3.125 \times (10^{-5})$ for UC1 and $th = 1.25 \times 10^{-6}$ for UC2 and UC3. Energy thresholds are computed using the iterative process proposed in [32].

Lastly, building on the findings of [15], which demonstrated that masked autoencoders achieve superior signal reconstruction in the frequency domain, we adopt a similar strategy in our approach. Specifically, we transform the raw acceleration data into its **spectrogram** representation, enabling the model to learn meaningful structural patterns through the reconstruction of masked spectrogram patches. Precisely, we feed the model with 100x100 time-frequency representations in the form of spectrograms. The first dimension corresponds to the temporal domain, while the second captures the frequency domain. While we notice that this window is big enough to obtain good task performance, we did not explore the downscaling of the spectrogram image to reduce the models' computational complexity, as the reduction of the input size has an impact only on the dimension of the last layer, whose size is negligible with respect to the whole model size.

For UC2 and UC3, we also need to generate the **target traffic variable**, while for UC1, samples are already separated between anomalous and normal based on time (see Sec. VI-A0a). Despite being generated differently for the two use cases, respectively from an optical camera and a WiM system, the target variables for UC2 and UC3 are identical. Namely, as described in [8], the target traffic load values for regression (y) in UC2 are set to:

$$y = \frac{\sum_{t=1}^{T}(l_t = k)}{10} \tag{1}$$

where $k = 1$ for light vehicles and $k = 2$ for heavy vehicles, and $\sum(l_t = k)$ indicates the number of samples that have label k in that window. Notice that y can also be a fractional number when the beginning/end of the window includes only a subset of a group of 10 consecutive samples sharing the same label. For UC3, the only difference is the lack of separation between light vehicles and heavy vehicles. After pre-processing, data are split between **train and test** as described in Section IV, and summarized in Table II.

Lastly, we also apply a post-processing smoothing technique on the output of our trained model for UC1. Specifically, we label a window as anomalous or normal based on the *median* over a variable size set of consecutive predictions. Namely, we experiment with median filters over $\{15, 30, 60, 120, 240\}$ consecutive windows.

B. Foundation Model Architecture

Our model architecture draws inspiration from the Masked Autoencoder proposed in [15], which was originally introduced to learn self-supervised representations from audio spectrograms. The model uses a stack of standard Transformer blocks for both the encoder and the decoder, each consisting of multiple attention layers. As a backbone, we use ViT-Base (ViT-B). The model takes as input acceleration spectrograms divided into non-overlapped regular grid patches, with a configurable portion of them being masked during training. The complete architecture is shown in Fig. 4.

Architectural modifications are necessary to fine-tune our foundation model based on the downstream tasks considered. Namely, the model architecture remains unchanged when fine-tuning on UC1, because the task is framed as an outlier detection [32]: during training, the model learns the latent representation of normal data only, so that during testing, the decoder is expected to fail in accurately reconstructing data that deviates from its learned patterns. Accordingly, those samples whose reconstruction error (computed as detailed in Sec. V-C1) exceeds a user-defined threshold are classified as anomalies or outliers. On the other hand, the decoder is

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

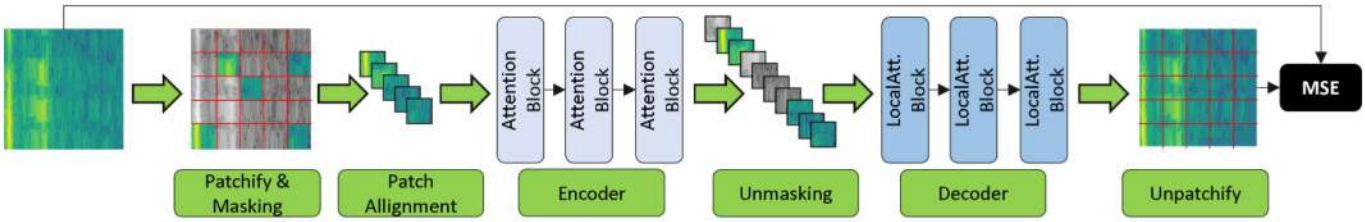


Fig. 4. Architecture of our foundation model transformer-based masked autoencoder during pre-training.

removed when fine-tuning on TLE tasks (UC2 and UC3), and a fully connected layer with a single neuron is appended directly to the latent representation produced by the encoder to produce the desired continuous traffic output defined in Eq. 1. The encoder is initialized with the weights learned during the self-supervised pre-training step, ensuring the transfer of the learned, general feature extraction. Additionally, to handle potential differences in input resolution, we interpolate the position embeddings as done in DeiT [40], preserving spatial consistency in learned representations.

C. Training Phases

The training of our masked autoencoder involves two main phases. Namely, the model is first pre-trained considering data from all three datasets and then fine-tuned on each single task. The two phases are detailed in the following, distinguishing the fine-tuning procedure for UC1 (AD) and for UC2/3 (TLE). In both phases, we use an AdamW optimizer [41] with different learning rates based on the training phase. For the pre-training phase, we use a learning rate of $0.25 \cdot 10^{-3}$, while for the fine-tuning phase a learning rate of $0.25 \cdot 10^{-2}$ for UC1, $0.25 \cdot 10^{-5}$ for UC2 and UC3, and a weight decay of 0.05. Following what was presented in [40], [42], we apply automatic gradient scaling to avoid exploding gradients. During pre-training, we also progressively increase the learning rate from 0 to the pre-training learning rate value over the first 100 epochs to improve the convergence. After this, the learning rate is reduced following the half-cyle of the cosine function. We pre-train our foundation model for 200 epochs with a batch size of 128, then fine-tune it for 400, 500, and 200 epochs with batches of 64, 8, and 128 samples on UC1, UC2, and UC3, respectively.

1) Pre-training: The full encoder-decoder model, with the architecture of Fig. 4 is initially pre-trained on a large amount of unlabeled data. We select pre-training data following the split described in Section II, combining the *training sets of all three UCs*. Note that, although our data have labels, we do not use them in this phase, to simulate a scenario where the costly process of labelling all available samples is not affordable.

During pre-training, the non-overlapping spectrogram patches of each input sample are masked with probability p (masking ratio) and aligned, before being passed to the transformer encoder. The latter then processes the non-masked patches to produce latent representations. Encoded patches are padded with trainable mask tokens and fed to the decoder, whose objective is to reconstruct the original spectrogram, including the missing parts. The encoder and decoder are jointly optimized to minimize the Mean Squared Error (MSE) between the reconstruction and the input spectrogram, averaged over masked patches. This pre-training is completely self-supervised and task-independent. Its objective is to learn robust latent representations of SHM data that allow the model to infer the underlying patterns in the data despite the missing information.

2) Fine-tuning: The details of the fine-tuning phase differ depending on the downstream task. For UC1 (AD), as mentioned above, we maintain the autoencoder structure of our model. Accordingly, the model is fine-tuned in an unsupervised way, using the same procedure just described, but considering only normal data from the UC1 dataset. At test time, we use a threshold on the reconstruction error to identify anomalies. For both our models and the baseline PCA, we follow a procedure similar to [32] to determine it. Namely, the threshold is initialized to the average MSE on non-anomalous training data, and then progressively increased by a small step (1/1000th of the average), until 100% specificity is achieved on a validation set, obtained as a 20% sample of the training dataset.

For UC2, the pre-trained model is fine-tuned on acceleration data and video recording on the viaduct presented in Sec. VI-A0b. Two separate fine-tunings are executed for light and heavy vehicle TLEs, starting from the same pre-trained model. For UC3, we consider the first half of the second day's data to fine-tune our foundation model, as described in VI-A0c. For these two downstream tasks, we follow a classic supervised fine-tuning procedure, where the model learns to predict the traffic load values rather than to reconstruct the input spectrograms. To fine-tune the encoder weights and train the added fully-connected layer, we use an initial lower learning rate of $0.25 \cdot 10^{-5}$ that is updated with a half-cycle cosine decay schedule. As a loss function, we use the MSE between the newly inserted fully connected layer's output prediction and the ground truth target variable.

D. Model Exploration and Knowledge Distillation

To select our model's hyperparameters, we do a search over the encoder/decoder depth and heads, the embedding dimensions, and the masking probability p of the masked autoencoder. To make this search manageable, we sample 20% of the pre-training data, further splitting them into 80% for training and 20% for validation. The task is the same as the pre-training, i.e., reconstructing the masked input data. The best configuration is selected based on the lowest error achieved on the validation set. We employ Bayesian

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

TABLE I
EXPLORED SPACE OF ARCHITECTURAL HYPERPARAMETERS FOR THE TRANSFORMER ENCODER-DECODER

Parameters	Values explored
Encoder depth	$[1, 3, 5, \mathbf{7}]$
# Encoder Heads	$[\mathbf{8}, 12, 16]$
Encoder embedding	$[1536, \mathbf{768}, 384, 192]$
Decoder depth	$[\mathbf{2}, 4, 6, 8]$
# Decoder Heads	$[8, \mathbf{16}, 24]$
Decoder embedding	$[1024, \mathbf{512}, 256, 128]$
Masking ratio	$[0, 0.2, 0.4, 0.6, \mathbf{0.8}]$

optimization with the open-source Optuna framework[1], fixing $p = 0$. Then, we explore p through a grid search with a step of 0.2. Table 1 summarizes the search space, with the optimal configuration highlighted in bold.

After this step, we also define a family of smaller models to explore the possibility of reducing the computational burden with minimal accuracy loss, targeting the deployment on sensor nodes with an extremely low memory budget. Starting from the frozen configuration of Table I, we systematically reduce the embedding dimensions of both the encoder and decoder, starting from the baseline configuration $(e_{dim}, d_{dim}) = (768, 512)$ and scaling them down by a factor of 2 until reaching $(e_{dim}, d_{dim}) = (24, 16)$. These new models are pre-trained on all datasets and subsequently fine-tuned on the three downstream tasks, following the same procedure described in the previous section. Additionally, to inherit our big model performance, we considered a modified fine-tuning procedure enhanced with Knowledge Distillation (KD). In this setup, considering the baseline model as a teacher, the student utilizes a multi-objective fine-tuning loss function composed of two contributions: the first one, $\mathcal{L}_{task}$, is the MAE computed on student prediction against the ground truth, while the second one, $\mathcal{L}_{KD}$, is the Root Mean Squared Error (RMSE) between student and teacher predictions. Through an empirical approach, we assessed that evenly balancing these two contributions leads to faster convergence of the overall loss, whose function becomes:

$$\mathcal{L} = 0.5 \cdot \mathcal{L}_{task}\left(y_s, y_{true}\right) + 0.5 \cdot \mathcal{L}_{KD}\left(y_s, y_t\right) \qquad (2)$$

where y_s, y_t, and y_{true} are the student, the teacher, and the true predictions, respectively. Even when using KD in fine-tuning, for all UCs, student models are always first pre-trained following the self-supervised procedure of Sec. V-C1.

VI. RESULTS

In this section, we first introduce the datasets used for training, validation and testing, summarized in Table II. Notice that, identically to state-of-the-art works, data splitting is tailored for each task, considering the fundamental differences between anomaly detection (UC1) and TLE (UC2 and UC3). Then, we introduce the comparison baselines and detail the results of our foundation models on the three use cases. We also quantify the impact of self-supervised pre-training on the performance of our models. Lastly, we analyze the trade-off between model size and performance and compare regular and KD-enhanced fine-tuning. In all experiments except the last one, we consider our largest masked autoencoder version, i.e., $(e_{dim}, d_{dim}) = (768, 512)$.

[1]https://optuna.org

A. Datasets

a) UC1: the employed dataset is the same as the one described in [32], comprising a total of 15 days of continuous monitoring of the viaduct. For our analysis, we consider only the central sensor of the chain, which is the one most influenced by the viaduct vibration. The data include measurements from 4 days before the maintenance intervention, labeled as anomalies, and two intervals of 4 and 7 days respectively, after the intervention, labeled as normal data. We select as the test set the two intervals of 4 days to have a perfect balance between anomalies and normal data. The remaining 7 days of normal data are used to train the AD algorithm (and the AD threshold). Note that anomalies are not used during training given that, for this task, we only consider unsupervised models trained to identify anomalies as outliers.

b) UC2: we analyze only section 10 of the viaduct, where an optical camera provides ground truth labels for TLE. This section includes seven sensors mounted on different tendons, but we use only the central one, which experiences the highest vibrations. The dataset consists of 31 minutes of recorded z-axis acceleration data and a synchronized video, split 70:30 for training (21 minutes and 42 seconds) and testing (9 minutes and 18 seconds). Data were collected between 8:00 a.m. and 9:00 a.m., with the camera operating at 10 FPS. Labels were assigned based on vehicle crossings, using the video: a value of 1 for light vehicles, 2 for heavy vehicles, and 0 otherwise. Given the accelerometer's 100 Hz sampling rate and the camera's 10 FPS, each label was assigned to a set of 10 consecutive acceleration samples. The final TLE label was computed as the sum of 1s and 2s in the window, divided by 10, following [8] (see Sec. V-A).

c) UC3: we utilize two days of data to train and evaluate our algorithms. We use the first complete day's data, spanning from 00.00 to 23.59, and covering a variety of traffic conditions only as pre-training data for the foundation model. We then use the first half of the following day (00.00 to 12.00) as additional training data for supervised fine-tuning on this specific task. Finally, the last half day's data is reserved for testing. It's important to note that, in the 600 m between the WiM and the bridge, vehicle speeds may vary, with potential lane changes, vehicle overtaking, and overlapping. As a result, vibrations on the bridge may not be entirely correlated with the data collected from the WiM, which can unavoidably lead to a loss of accuracy in this analysis.

B. Comparison Baselines

We use different baseline algorithms depending on the UC, given the difference in the solved task, i.e., AD vs. TLE. For UC1, we compare against the top-performing model of [32], a PCA that considers a window dimension of 5 seconds (500 samples at 100 Hz) and a post-processing median filter

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

TABLE II
INFORMATION ABOUT THE USE CASES' DATASET. ABBREVIATIONS: AD: ANOMALY DETECTION, TLE: TRAFFIC LOAD ESTIMATION.

	Task	N. of Samples [Train / Test]	Time [Train / Test]	Data Dimension	Data Stride	N.of Sens.	Label
UC1	AD	302.4k / (172.8k N., 172.8k A.)	7d / 8d (4d N., 4d A.)	5s × 100Hz	2s × 100Hz	1	Normal / Anomaly
UC2	TLE	651 / 279	21:42 / 9:18 min:sec	60s × 100Hz	2s × 100Hz	1	# of Light/Heavy Vehicles
UC3	TLE	699.9k / 50k	36 hours / 6 hours	60s × 100Hz	15s × 100Hz	142	# of Vehicles

with varying length to aggregate consecutive predictions. We underline that this solution has already been shown to outperform convolutional and fully connected deep autoencoders, as demonstrated in [32].

For UC2 and UC3, we compare with the five supervised algorithms proposed in [8], i.e., the work that first introduced the UC2 dataset. It is worth noticing that this is the task with the largest dataset available for training: this could further explain the superiority of foundation-model-based approaches, which are expected to thrive when more data are available, and, at the same time, justify the limits of SoA approaches, as their shallow nature may be unsuited to handle this complexity. Namely, we consider: i) a K-Nearest Neighbors (k-NN); ii) a Linear Regressor (LR); iii) a Random Forest (RF) trained with the variance reduction criterion [43]; iv) a Support Vector Regressor (SVR); v) a Multi-Layer Perceptron (MLP) with ReLU activations, trained with the *Adam* optimizer, a learning rate of 10^{-3} and a mini-batch size of 200. In addition to that, we consider two alternative deep learning architectures, a Temporal Convolutional Network (TCN) and a Long Short-Term Memory (LSTM) network. We implement the TCN architecture of [44] composed of four 1D convolutional layers with increasing dilation factor from first to last block, followed by a regression head consisting of two linear layers with ReLU activation and dropout. For the LSTM, we consider a variation of the architecture from [45] reducing its layers from 4 to 3 due to overfitting, and changing the hidden size to match the one of our Transformer Encoder (768). Also in this case, the regression head is composed of two linear layers with ReLU activation and dropout. On these last two architectures, we test both direct supervised training on the task-specific dataset, as described in Section V-C, and a two-phase foundation model approach including self-supervised pre-training on all the unlabeled data followed by supervised fine-tuning. For the latter, we use the same masked autoencoder framework of Figure 4, except that the Transformer-based Encoder and Decoder are replaced by TCN and LSTM respectively. We build the Decoders architectures symmetrically to the Encoders: for the TCN, we replace convolutions with up-sampling convolutions, while for the LSTM, we add a linear layer to project the output of the encoder to the decoder embedding dimension (512), once again matched to the one of our Transformer. Section VI-D only reports the results with pre-training, which has proven superior for both types of model. A summary of the baseline models and their hyperparameters is shown in Table III.

TABLE III
MODELS EMPLOYED IN OUR EXPERIMENTS.

Model	Task	Hyperparameters
State-of-the-art		
LR	TLE	-
RF	TLE	Depth = 200, Trees = 30
k-NN	TLE	k = 7
MLP	TLE	Layers = 3, Neurons = 100
SVR	TLE	Kernel = RBF, C = 10.0, ϵ=0.1
TCN	TLE	4 1D-conv, 2 Linear (See Section VI-B)
LSTM	TLE	3 LSTMs, 2 Linear (See Section VI-B)
PCA	AD	CF = 32
Our Work		
SHM FoMo	TLE, AD	See Section V

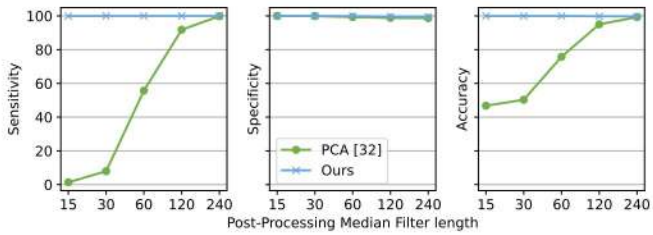


Fig. 5. SoA PCA [32] vs. masked autoencoder performance while varying the smoothing post-processing median filter length.

C. Anomaly Detection (UC1) Results

We evaluate AD models in terms of *specificity*, computed as the proportion of correctly classified normal windows, *sensitivity*, i.e., the proportion of correctly classified anomalous windows, and *accuracy*, i.e., the overall ratio of correctly classified windows, providing a general measure of performance. Figure 5 shows the comparative performance of PCA and our foundation model while varying the number of windows outputs fed to the median filter as post-processing. The graph highlights the increasing trend of accuracy, sensitivity, and specificity as the dimension increases for both methods. However, we can observe a significant improvement in all evaluation metrics of our model compared to the PCA, especially for short time windows. For example, when applying the median smoothing to 60 windows, our masked autoencoder achieves an accuracy of 99.92%, while the PCA only reaches 75.76%. To match the accuracy of our model, the PCA needs a post-processing median filter of 240 windows, i.e., 4x larger. The sensitivity shows a similar trend: for a 60-windows smoothing, our model reaches 99.9%, compared to 55.68% of the PCA.

Having a shorter post-processing median filter translates into a significantly lower anomaly detection latency. For example, given that each window is 5s long, a median filter over 240 windows, as required by the PCA, corresponds to a latency of 20 minutes before an anomaly can be signalled. In contrast, our method reduces this latency to 5 minutes.

Additionally, the specificity values demonstrate the masked autoencoder's superiority in correctly identifying normal data

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

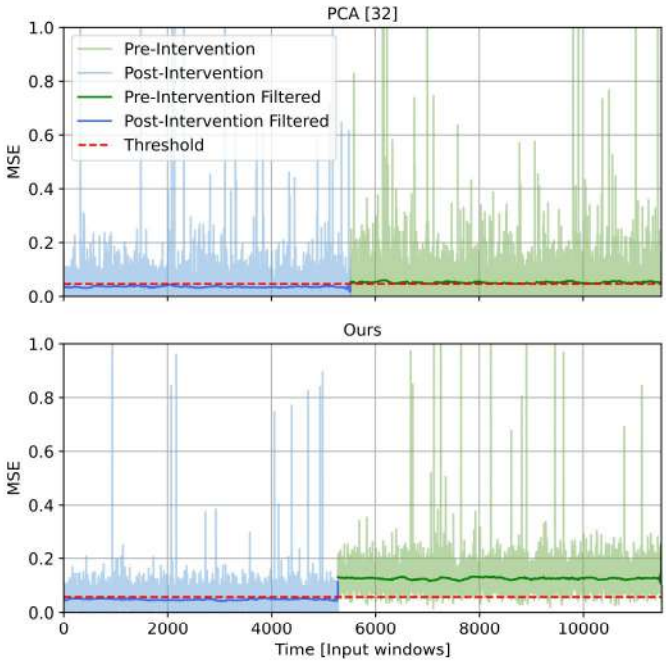


Fig. 6. Reconstruction error of the PCA [32] and our proposed model, for normal and anomalous windows

instances. We achieve an almost perfect specificity (min. 99.97%) for all time window dimensions, whereas PCA shows lower specificity values, ranging from 96.21% to 99.87%.

To visualize the performance of our model compared to the PCA, Figure 6 shows the reconstruction error (MSE) of normal and anomalous windows. The darker line shows the output of median-smoothing over 120 windows. From the figure, we can notice that the PCA reconstruction error is more noisy, and, after the smoothing post-processing, it shows a very small margin between normal and anomalous signals. On the other hand, our approach shows a much more pronounced difference between pre and post-intervention data, thus being able to more robustly tell apart normal and anomalous windows.

D. Traffic Load Estimation (UC2 and UC3) Results

To compare the performance of TLE models, we consider five different metrics: *Mean Squared Error (MSE)*, which measures the average squared difference between predicted and true values; *Mean Absolute Error (MAE)*, which computes the average absolute difference between predictions and ground truth; R^2 *score*, which quantifies how well the model explains the variance in the data, with values closer to 1 indicating better performance; *Mean Squared Percentage Error (MSE%)*, and *Mean Absolute Percentage Error (MAE%)*, obtained respectively by dividing MSE and MAE by the average predicted value on the test set, allowing for a relative error assessment with respect to the true values.

1) UC2: Table IV reports the results in terms of these metrics obtained on UC2 for light and heavy vehicles. Considering the latter, we outperform the best baseline models from [8] (SVR and RF) in all metrics. However, our foundation model reduces the MSE by 3.2x, the MAE by 1.3x, and improves

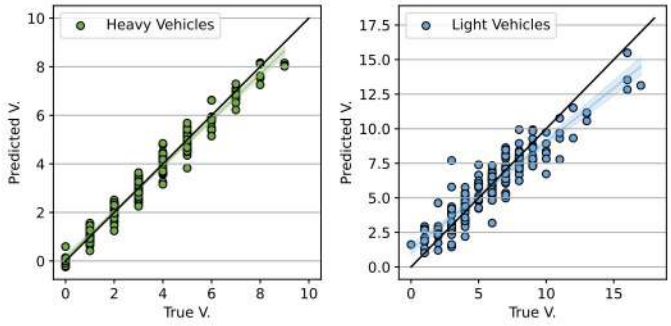


Fig. 7. True vs predicted values of the TLE regression task for heavy and light vehicles. The black line represents the ideal prediction (true value = predicted value). The coloured straight lines are the linear interpolation of the predicted samples, and the light-coloured area corresponds to the 99.5% confidence interval of the interpolation.

the R^2 of +0.06. We impute this to the increased capacity of our foundation model in combination with the transformer architecture, which permits to better leverage the non-linear and higher-order interactions within acceleration data that correlate with traffic flow, while current SoA machine learning methods exhibit limitations in capturing these intricate relationships. Noteworthy, the other deep learning architectures tested do not attain satisfactory performance, being inferior to classical machine learning algorithms Additionally, notice how there is no single best SoA model, while our approach convincingly overcomes them all in every considered metric. For the prediction of the light vehicle traffic, our foundation model outperforms the SoA in terms of MSE (1.6x reduction) and R^2 (+0.06) and is the second best in terms of MAE (1.03x increase over the best baseline model, the RF). Importantly, having a significantly lower MSE and a slightly higher MAE means that our approach does not output predictions that are far off the ground truth, as the quadratic function of the MSE penalizes major mistakes. Thus, it could be more amenable for a real-world deployment than the baseline RF.

Figure 7 plots the ground truth target variable against our model's predictions to further highlight our approach's effectiveness. As shown, most predictions are close to the target without noticeable outliers. The black line represents the ideal prediction, where true values exactly correspond to the predicted values. The coloured straight lines correspond to the linear interpolation of the predicted samples, while the light-coloured area corresponds to the 99.5% confidence interval of the interpolation. For the heavy vehicle case, the confidence interval almost overlaps with the ideal prediction across traffic conditions (i.e., x-axis values), further validating the effectiveness of our approach in this scenario. For light vehicles, when facing high traffic, the model's confidence interval slightly deviates from the ideal prediction, underestimating the traffic on the viaduct. This may be a consequence of the lower weight of light vehicles. The latter generate vibrations with lower amplitudes on the viaduct, where vibrations caused by multiple vehicles can be easily mistaken for a single vehicle in light traffic. Further, the model is biased towards lower-value TLE estimations, given the scarcity of high-traffic window samples.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

TABLE IV
COMPARISON BETWEEN THE ALGORITHMS OF [8] AND THE MASKED AUTOENCODER PROPOSED IN OUR WORK ON UC2. THE COMPARISON USES THE FEATURES EXTRACTION OF [8] FOR THE BASELINES AND RAW DATA FOR OUR METHOD. BOLD = BEST RESULT, UNDERLINE = SECOND BEST.

	Heavy Vehicles					Light Vehicles				
Model	MSE	MAE	R^2	MSE%	MAE%	MSE	MAE	R^2	MSE%	MAE%
State-of-the-art										
SVR	0.35	0.42	0.91	10.76	12.85	2.23	1.05	0.76	37.03	17.50
RF	0.38	0.32	0.91	11.76	9.66	1.50	**0.73**	0.84	25.02	**12.10**
MLP	0.53	0.53	0.87	16.37	16.37	2.99	1.35	0.67	49.69	22.44
kNN	0.37	0.41	0.91	11.47	12.45	2.26	1.09	0.75	37.59	18.14
LR	1.19	0.84	0.71	36.57	25.82	7.34	2.14	0.20	121.94	35.67
TCN	2.45	1.28	0.40	75.10	39.31	8.96	2.30	0.02	148.73	38.26
LSTM	1.50	0.95	0.63	45.99	29.16	9.48	2.41	-0.03	157.36	40.01
Our Work										
SHM FoMo	**0.11**	**0.25**	**0.97**	**3.42**	**7.88**	**0.95**	0.75	**0.90**	**15.92**	12.44

TABLE V
COMPARISON BETWEEN THE ALGORITHMS OF [8] AND THE MASKED AUTOENCODER PROPOSED IN OUR WORK ON THE UC3. THE COMPARISON USES ALL THE EXTRACTED FEATURES FOR THE BASELINES AND RAW DATA FOR THE MASKED AUTOENCODER.

Model	MSE	MAE	R^2	MSE%	MAE%
State-of-the-art					
LR	20.53	2.05	-14.40	1091.02	109.03
RF	2.23	1.10	-0.67	118.48	58.30
kNN	1.34	0.90	-0.01	71.14	47.81
MLP	1.20	0.85	0.10	63.89	45.39
SVR	1.39	0.91	-0.04	73.77	48.19
TCN	1.69	1.06	-0.27	89.98	56.17
LSTM	0.65	0.59	0.51	34.36	31.15
Our Work					
SHM FoMo	**0.62**	**0.57**	**0.54**	**32.93**	**30.13**

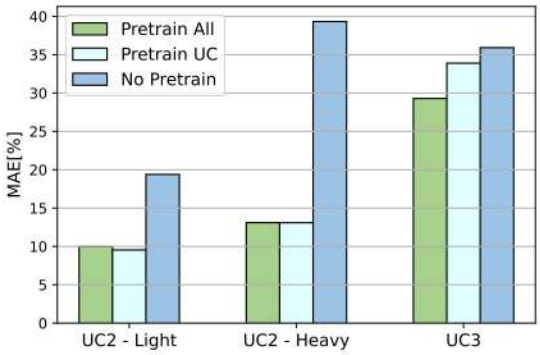


Fig. 8. Impact of different pre-trainings on masked autoencoder outcome.

2) UC3: The results of UC3 are shown in Table V. We compare our approach with the same models considered for UC2. However, since there is no previous literature on this dataset, we re-train all baseline models with the same data split used for our foundation model while keeping the same hyperparameter settings of [8]. Results show that our foundation model overcomes state-of-the-art performance on all metrics, improving the MSE and MAE of the second-best, the LSTM, by 0.03 and 0.02, respectively, while increasing the R^2 score by 0.03 as well. We observe two general trends: first, the performance of UC3 is worse compared to UC2's on every metric and for every algorithm. Second, the second-best performing model is the LSTM, trained with the same foundation-model approach as our transformer. This is in strong contrast with UC2, where both the other deep learning based models, TCN and LSTM, reached low performance.

The observed performance discrepancy for both state-of-the-art and our proposed approaches on UC3 compared to UC2 (and their performance ranking) can be attributed to two key factors. First, as anticipated in Section IV, the label assignment procedure likely introduces noise into the training data. The physical separation of the WiM system from the bridge affects the correlation between bridge vibrations and WiM collected data. This misalignment between predicted and actual bridge conditions may partially explain the performance drop observed in this use case. Sensor placement may also affect task difficulty. In fact, unlike UC2, sensors are positioned on two external beams per span, which might lead to reduced sensitivity to vehicle-induced vibrations. Secondly, UC3 comprises roughly 750,000 samples, over three orders of magnitude more than UC2's dataset, 1,000 samples. DNN models benefit more than classic ML ones from large-scale data, reaching, as in the case of the LSTM, significantly better performance compared to UC2.

E. Impact of pre-training

In this section, we demonstrate that self-supervised pre-training is a fundamental contributor to the performance of our foundation models. Considering UC2 and 3, we apply three different training setups:

- **No Pretrain**: the pre-training step is completely disregarded, and the model is directly trained in a supervised way considering only UC-specific data;
- **Pretrain UC**: the model is first pre-trained and then fine-tuned only considering data of the specific use case. Thus, the *same training data* are applied during the masked autoencoder self-supervised pre-training training and during supervised fine-tuning (after replacing the decoder with the classification head);
- **Pretrain All**: the model is pre-trained on all three datasets (combining the respective training sets) and fine-tuned on UC-specific data.

The test MAE% obtained by our foundation model in the three cases are reported in Figure 8. As shown, *No Pretrain* always

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

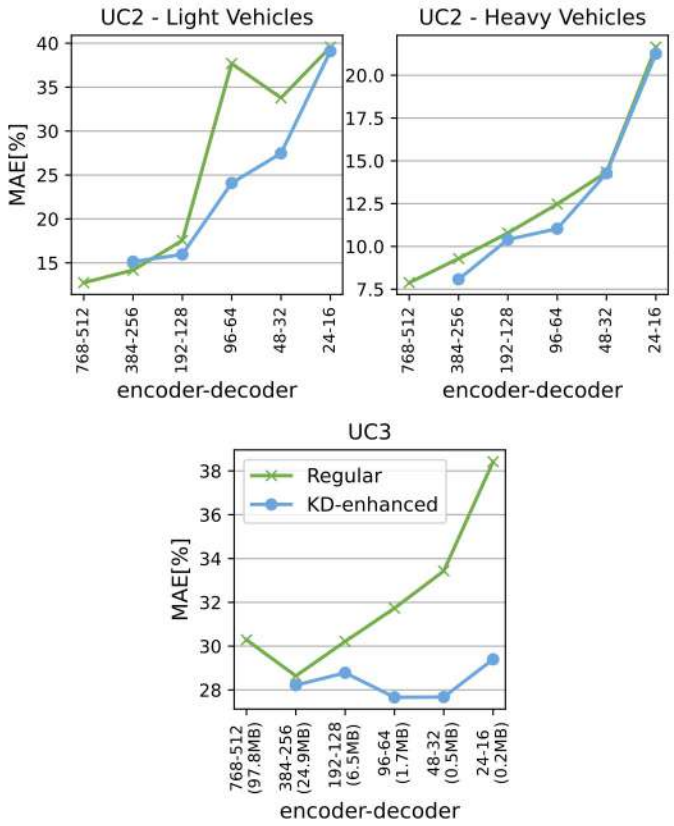


Fig. 9. MAE% comparison between regular and KD-enhanced fine-tuning. All graphs share the same x-axis.

performs worse than the other two setups, underlying the usefulness of self-supervised pre-training. Quantitatively, *Pretrain All* reduces the MAE% by 1.94×, 3× and 1.2× on UC2-Light, UC2-Heavy and UC3, respectively, compared to *No Pretrain*. As expected, the benefits of pre-training are more evident for UC2, which has the smallest dataset. Moreover, pre-training the foundation model on all three datasets shows the best results for UC3 (with *Pretrain All* achieving 1.2× lower MAE% compared to *Pretrain UC*), while no clear advantage is evident for UC2. This is in accordance with the observation of the previous section that UC3 is the most difficult downstream tasks among the three, thus benefitting more from high-quality and generalizable representations. These findings indicate that the effectiveness of this approach might be further amplified by leveraging substantially larger and more diverse datasets for pre-training.

F. Model size optimization and Knowledge Distillation

We also analyze the trade-off between model size and performance. For all UCs, following the findings of the previous section, we use a *Pretrain All* scheme, and we compare how regular supervised fine-tuning stands against KD-enhanced fine-tuning. We focus on TLE tasks because, for UC1, we were able to obtain the same performance shown in Figure 5 for the largest transformer, with a model of just 0.2MB, with $(e_{dim}, d_{dim}) = (24, 16)$. Results for UC2 and UC3 are shown in Figure 9. For clarity's sake, we present results only in terms of MAE%, as other metrics follow similar trends. As shown, KD helps improve the performance of the models in almost all configurations, with the exception of a few cases related to UC2, especially in the low-size regime, where it provides marginal advantages. This is motivated by the fact that student networks that are too small cannot imitate the predictions of much larger teachers. Hence, the KD loss only adds noise to the training signal. At most, on UC2, KD improves the MAE by 1.6× on light vehicles, at (e_{dim}, d_{dim}) =(96, 64), and by 1.15× on heavy vehicles, at (e_{dim}, d_{dim}) =(384, 256), compared to regular fine-tuning. In the latter case, the KD-enhanced model reaches a MAE% that is just 1.2× higher than the teacher while being 3.9× smaller.

The most interesting results are obtained on UC3. Here, we first note that reducing the model size to (e_{dim}, d_{dim}) =(384-256) is beneficial for performance (5% reduction for a 3.9× smaller size) even without KD. Moreover, we can achieve very competitive results even with smaller models thanks to our KD-enhanced fine-tuning setup, which is very effective. The most interesting result is achieved at $(e_{dim}, d_{dim}) = (48, 32)$, which further reduces the MAE% by 8.6% compared to the baseline, with a model whose parameters only require 0.5MB, i.e., 195× smaller than the original one.

These results show the effectiveness of KD in reducing model size while preserving as much as possible the performance of foundation models from the perspective of real-world deployment resource-constrained devices. The two smallest student configurations require less than 0.7 MBs and could be even deployed on the data collection nodes mentioned in Section IV, which are equipped with 1 MB of Flash memory. However, it must be underlined that, on UC2, these downsized models remain hindered by performance gaps that might be too large depending on application-specific requirements.

G. Deployment results

While further research is necessary to port these models on SHM sensor nodes, in this section we show that running them *in real-time* on an edge gateway (present in virtually every sensor node installation, including the ones we considered) is feasible without any loss in accuracy. Table VI summarizes the deployment results of our architectures on an NVIDIA Jetson Nano, as a proxy for a typical AI-oriented SHM gateway. The module features an NVIDIA Maxwell architecture with 128 CUDA cores and a Quad-core ARM Cortex-A57 MPCore processor, equipped with 4 GB of 64-bit LPDDR4 memory and 16 GB of storage. Inference times were measured across all architectures to assess their real-time feasibility in deployment scenarios, averaging results across 1000 test samples using PyTorch 1.1.0, with model and data in full precision (fp32); because the deployed models are identical to those evaluated offline, their accuracy remains unchanged.

Among all the use cases, the biggest transformer model is the one for UC1 (which includes the Decoder), with 36M parameters: in fp32, this model occupies a total of 144 MB, easily fitting the target gateway's memory. This model achieves an inference latency of 169.36 ms. While the PCA-based anomaly detector is faster, our transformer is still well

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

TABLE VI
DEPLOYMENT RESULTS OF FOR A SINGLE INFERENCE STEP ON NVIDIA JETSON NANO. UC2/UC3 RESULTS CONSIDER A 60 S INPUT WINDOW.

Task	UC1		UC2/UC3								
Model	PCA	Ours	LR	RF	k-NN	MLP	SVR	TCN	LSTM	Ours	Ours (distilled)
Params	$<$ 100k	36 M	$<$ 100k	$<$ 100k	$<$ 100k	$<$ 100k	$<$ 100k	11 M	42 M	25 M	130k
Latency [ms]	14.81	169.36	0.26	0.72	4.63	0.66	0.52	127.91	1533.26	164.93	40.12

below the 2-second constraint imposed by the stride between successive input windows.

For UC2 and UC3, both using the same model where the Decoder has been replaced by a fully connected layer, we achieve 164.93 ms latency, well below the real time constraints of 2 / 15 seconds of UC2 / UC3. As expected, the distilled model is significantly faster, requiring only 40 ms. Classical ML baselines exhibit latencies below 1 ms, which are however unnecessary in this context.

VII. CONCLUSIONS

We proposed a task-independent foundation model for SHM applications, which can achieve state-of-the-art performance on two tasks, Anomaly Detection and Traffic Load Estimation, considering three different datasets. Our transformer-based masked autoencoder learns powerful representations of vibration data gathered by accelerometers during the pre-training step and can be then effectively fine-tuned on the specific downstream task. Even if limited to a few SHM tasks, we believe that our work points to a new and exciting direction in ML for SHM research. Future works will include using larger and more diverse pre-training datasets, as our results already suggested this could lead to possible benefits, investigating more SHM tasks involving different civil infrastructures, as well as experimenting more thoroughly with model hyperparameters.

VIII. ACKNOWLEDGMENT

This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 117/2023. We thank Sacertis Ingegneria Srl for providing the data for this research.

REFERENCES

[1] G. T. Webb *et al.*, "Analysis of fiber-optic strain-monitoring data from a prestressed concrete bridge," *Journal of Bridge Engineering*, vol. 22, no. 5, p. 05017002, 2017.

[2] E. Odat *et al.*, "Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 5, pp. 1593–1606, 2017.

[3] C.-H. Lin *et al.*, "A real-time bridge structural health monitoring device using cost-effective one-axis accelerometers," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2015, pp. 1–2.

[4] A. Hashad, "Using dynamic analysis of site vibration to select the suitable vibration limit," *HBRC Journal*, vol. 14, no. 2, pp. 180–188, 2018.

[5] S. Kamkar and R. Safabakhsh, "Vehicle detection, counting and classification in various conditions," *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 406–413, 2016.

[6] V. Shahsavari *et al.*, "Diagnostic testing of a vertical lift truss bridge for model verification and decision-making support," *Frontiers in Built Environment*, vol. 5, 2019.

[7] E. Parisi *et al.*, "Time and frequency domain assessment of low-power mems accelerometers for structural health monitoring," in *2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*, 2022, pp. 234–239.

[8] A. Burrello *et al.*, "Traffic load estimation from structural health monitoring sensors using supervised learning," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100704, 2022.

[9] R. P. Finotti *et al.*, "An shm approach using machine learning and statistical indicators extracted from raw dynamic measurements," *Latin American Journal of Solids and Structures*, 2019.

[10] S.-Y. Kim and M. Mukhiddinov, "Data anomaly detection for structural health monitoring based on a convolutional neural network," *Sensors*, vol. 23, p. 8525, 2023.

[11] K. Yang *et al.*, "Unsupervised long-term damage detection in an uncontrolled environment through optimal autoencoder," *Mechanical Systems and Signal Processing*, vol. 199, p. 110473, 2023.

[12] E. M. Coraça *et al.*, "An unsupervised structural health monitoring framework based on variational autoencoders and hidden markov models," *Reliability Engineering & System Safety*, vol. 231, p. 109025, 2023.

[13] R. Wang *et al.*, "A novel transformer-based semantic segmentation framework for structural condition assessment," *Structural Health Monitoring*, vol. 23, no. 2, pp. 1170–1183, 2024.

[14] R. Bommasani, P. Liang *et al.*, "On the opportunities and risks of foundation models," *ArXiv*, 2021.

[15] P.-Y. Huang *et al.*, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[16] Y. Yu *et al.*, "State-of-the-art review on bridge weigh-in-motion technology," *Advances in Structural Engineering*, vol. 19, no. 9, pp. 1514–1530, 2016.

[17] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] E. Wang *et al.*, "Multi-modal knowledge graphs representation learning via multi-headed self-attention," *Information Fusion*, vol. 88, pp. 78–85, 2022.

[19] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, 2021.

[20] K. He *et al.*, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 979–15 988.

[21] G. Paaß and S. Giesselbach, "Foundation models for natural language processing – pre-trained language models integrating media," *ArXiv*, 2023.

[22] L. Yuan *et al.*, "Florence: A new foundation model for computer vision," *ArXiv*, 2021.

[23] X. Sun *et al.*, "Dime-fm : Distilling multimodal and efficient foundation models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 15 475–15 487.

[24] J. A. Miller *et al.*, "A survey of deep learning and foundation models for time series forecasting," *ArXiv*, 2024.

[25] M. Goswami *et al.*, "Moment: A family of open time-series foundation models," *ArXiv*, 2024.

[26] D. Lydon *et al.*, "Use of a roving computer vision system to compare anomaly detection techniques for health monitoring of bridges," *Journal of Civil Structural Health Monitoring*, vol. 12, no. 6, pp. 1299–1316, 2022.

[27] O. Abdeljaber *et al.*, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *Journal of Sound and Vibration*, vol. 388, pp. 154–170, 2017.

[28] L. Zanatta *et al.*, "Damage detection in structural health monitoring with spiking neural networks," in *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT)*, 2021, pp. 105–110.

[29] F. Barchi *et al.*, "Spiking neural network-based near-sensor computing for damage detection in structural health monitoring," *Future Internet*, vol. 13, no. 8, p. 219, 2021.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097

[30] J. Liu *et al.*, "Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction," *Mechanical Systems and Signal Processing*, vol. 136, p. 106454, 2020.
[31] K. A. Eltouny and X. Liang, "Large-scale structural health monitoring using composite recurrent neural networks and grid environments," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 3, pp. 271–287, 2023.
[32] A. Moallemi *et al.*, "Exploring scalable, distributed real-time anomaly detection for bridge health monitoring," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 660–17 674, 2022.
[33] H. Dong *et al.*, "Improved robust vehicle detection and identification based on single magnetic sensor," *Ieee Access*, vol. 6, pp. 5247–5255, 2018.
[34] Q. Wang *et al.*, "Roadside magnetic sensor system for vehicle detection in urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1365–1374, 2017.
[35] A. Boukerche, D. Zhong, and P. Sun, "Feco: An efficient deep reinforcement learning-based fuel-economic traffic signal control scheme," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 1, pp. 144–156, 2022.
[36] A. Ferguson *et al.*, "Detecting vehicle loading events in bridge rotation data measured with multi-axial accelerometers," *Sensors*, vol. 22, p. 4994, 2022.
[37] A. Burrello *et al.*, "Enhancing structural health monitoring with vehicle identification and tracking," in *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2020, pp. 1–6.
[38] "lis344alh sensor, STMicroelectronics."
[39] "Hts221 sensor, STMicroelectronics."
[40] H. Touvron *et al.*, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*, vol. 139, 2021, pp. 10 347–10 357.
[41] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *ArXiv*, 2019.
[42] H. Bao *et al.*, "BEit: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022.
[43] L. Breiman *et al.*, *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
[44] S. Bai *et al.*, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *ArXiv*, 2018.
[45] P. Ba *et al.*, "Structural monitoring data repair based on a long short-term memory neural network," *Scientific Reports*, vol. 14, 2024.



Luca Benfenati received his B.Sc. in Electronic and Telecommunications Engineering at University of Bologna, Italy and M.Sc. in Data Science and Engineering at the Politecnico of Turin, Italy, in 2020 and 2023, respectively. He is currently a Ph.D. student in Computer Engineering at Politecnico di Torino. His research interests include large language model inference optimization, transformer-based approaches and foundation models.



Daniele Jahier Pagliari received the M.Sc. and Ph.D. degrees in computer engineering from the Politecnico di Torino, Turin, Italy, in 2014 and 2018, respectively. He is currently an Assistant Professor with the Politecnico di Torino. His research interests are in the computer-aided design and optimization of digital circuits and systems, with a particular focus on energy-efficiency aspects and on emerging applications, such as machine learning at the edge.



Luca Zanatta received the M.Sc. degree in Biomedical Engineering from the Electronics and Telecommunications Department (DET) of Politecnico di Torino in 2019. Since October 2020, he is pursuing a Ph.D. at the ECS Lab of the Alma Mater Studiorum-Università di Bologna, Bologna, Italy. His research focuses on Spiking Neural Networks (SNN) and Reinforcement Learning.



Yhorman Alexander Bedoya Velez received his B.Sc. in Industrial Engineering from Universidad de Antioquia, Medellín, Colombia, in 2023, followed by an M.Sc. in Data Science and Engineering from Politecnico di Torino, Turin, Italy, also in 2023. He currently serves as a Data Scientist specializing in Supply Chain Optimization in the private sector



Andrea Acquaviva (Ph.D.) received the Ph.D. degree in Electrical Engineering in 2003 and in Complex Systems for Life Science in 2018. He was a research intern at HP Labs Palo Alto, CA, USA (2001–2003) and a visiting researcher at LSI lab, EPFL, CH (2007–2008). He has been the coordinator of various research projects at European and National levels in the field of embedded software, IoT and smart city/energy/buildings. His research focuses on: (i) compilers and operating systems for heterogeneous embedded platforms, multicores, Internet-of-Things (IoT) devices; (ii) Embedded software for Cyber–Physical Systems (CPS); (iii) Neuromorphic and brain inspired computing. He also carries on research on computational aspects of complex biological systems. Researches by Prof. Andrea Acquaviva (between 2000 and 2022) yielded around 300 papers in international journals and peer-reviewed international conference proceedings.



Massimo Poncino (Fellow, IEEE) received the Dr.Eng. degree in electrical engineering and the Ph.D. degree in computer engineering from Politecnico di Torino, Torino, Italy, in 1989 and 1993, respectively. He is a Full Professor of Computer Engineering with Politecnico di Torino. His current research interests include various aspects of design automation of digital systems, with emphasis on the modeling and optimization of energy-efficient systems.



Enrico Macii (Fellow, IEEE) received the Laurea degree in electrical engineering from Politecnico di Torino, Torino, Italy, in 1990, the Laurea degree in computer science from the Università di Torino, Turin, Italy, in 1991, and the Ph.D. degree in computer engineering from Politecnico di Torino in 1995. He is a Full Professor of Computer Engineering with Politecnico di Torino. His research interests are in the design of digital electronic circuits and systems, with a particular emphasis on low-power consumption aspects energy efficiency, sustainable urban mobility, clean, and intelligent manufacturing

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Sustainable Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2025.3592097



Luca Benini holds the chair of digital Circuits and systems at ETHZ and is Full Professor at the University of Bologna. He received a PhD from Stanford University. He served as chief architect in STmicroelectronics France. Dr. Benini's research interests are in energy-efficient parallel computing systems, smart sensing micro-systems and machine learning hardware. He has published more than 1000 peer-reviewed papers and five books. He is a fellow of the ACM and a member of the Academia Europaea



Alessio Burrello received his M.Sc. and Ph.D degrees in Electronic Engineering at Politecnico of Turin and University of Bologna, respectively, in 2018 and 2023. He is currently a research assistant at Politecnico di Torino. His research interests include parallel programming models for embedded systems, machine and deep learning, hardware-oriented deep learning, and code optimization for multi-core systems.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on October 03,2025 at 14:50:50 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.