A Tiny Convolutional Neural Network driven by System Identification for Vibration Anomaly Detection at the Extreme Edge

This is the final peer-reviewed author’s accepted manuscript (postprint) of the following publication:

Published Version:

Availability:
This version is available at: https://hdl.handle.net/11585/937833 since: 2023-08-03

Published:
DOI: http://doi.org/10.1109/COINS57856.2023.10189195

Terms of use:
Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher’s website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/). When citing, please refer to the published version.

(Article begins on next page)
A Tiny Convolutional Neural Network driven by System Identification for Vibration Anomaly Detection at the Extreme Edge

Federica Zonzini
DEI - Electrical, Electronic, and Information Engineering
University of Bologna
40136 Bologna, Italy
federica.zonzini@unibo.it

Letizia Burioli
DEI - Electrical, Electronic, and Information Engineering
University of Bologna
40136 Bologna, Italy
letizia.burioli@studio.unibo.it

Andi Gashi
DEI - Electrical, Electronic, and Information Engineering
University of Bologna
40136 Bologna, Italy
andi.gashi@studio.unibo.it

Nicola Francesco Mancini
DEI - Electrical, Electronic, and Information Engineering
University of Bologna
40136 Bologna, Italy
nicola.mancini11@studio.unibo.it

Luca De Marchi
DEI - Electrical, Electronic, and Information Engineering
University of Bologna
40136 Bologna, Italy
l.demarchi@unibo.it

Abstract—Vibration data analysis is the driving tool for the Structural Health Monitoring (SHM) of structures in the dynamic regime, i.e., structures showing important oscillatory behaviours, which largely dominate the transportation backbone: from terrestrial/aerial vehicles (e.g., trains, aircraft, etc.) to the supporting infrastructures (e.g., bridges, viaducts, etc.). Outstanding opportunities have recently been disclosed in the field of Intelligent Transportation Systems (ITS) by the advent of sensor-near processing functionalities, eventually empowered by Artificial Intelligence (AI). The latter allow for the extraction of damage-sensitive features at the extreme edge, without the need of transmitting long time series over the monitoring network. In this work, we explore for the first time a novel anomaly detection workflow for on-sensor vibration diagnostics, which combines the unique advantages of embedded System Identification (eSysId) as a data compression strategy with the computational/energy advantages of Tiny Machine Learning (TinyML). Experimental results conducted on a representative SHM dataset demonstrate that the proposed pipeline can achieve high classification scores (above 90%) for the health assessment of the well-known Z24 bridge. In particular, the minimal inference time (less than 44 ms) and power consumption performed while running on three different general-purpose microprocessors make it a promising solution for the development of the next generation of SHM-oriented ITS.

I. INTRODUCTION

Among the many challenges faced by the Internet-of-Things society, the one related to data outsourcing deserves primary attention due to the huge power consumption of the transmission module, which poorly fits the constrained energy budget admitted for the realization of long-lasting battery-operated systems [1]. Structural Health Monitoring (SHM) perfectly endorses this requirement, since the adoption of very dense wireless networked solutions has become predominantly in the last years [2]. Moreover, this need for autonomous sensing devices is even exacerbated in transportation scenarios by considering that sensors might be installed in physically inaccessible locations (e.g., under a bridge deck, integrated in railway mechanical components, embedded in aircraft or aerial vehicle wings, etc.); this poses relevant constraints in terms of safety of the maintenance operators in case repair actions are urged, and represents one of the main reasons for the development of intelligent transportation systems (ITS) and facilities.

Given these premises, a current research trend in SHM-ITS tackles the optimization of the power transmission from a data science perspective, i.e., it aims at exploiting the on-board feature extraction capabilities made available by smart
sensors to extract semantic information in strict proximity to
the sampling point. As a fundamental byproduct of this sensor-
near perspective, a small batch of processed features can be
outsourced in place of raw time series, with considerable
reduction in the length of the data payload to be delivered over
the monitoring network. To this purpose, data compression
methods are usually employed by taking advantage of the
specific properties of the class of processed signals. This is the
case of vibration-based diagnostics [3], which aims at moni-
toring assets in dynamic regime by tracking the evolution over
time of few peak spectral values, called natural frequencies,
which capture most of the total structural energy.

Nevertheless, two obstacles are currently limiting the
widest porting of advanced data reduction strategies at
the extreme edge. Firstly, it is worth mentioning the huge
computational complexity involved by conventional algorithms
and the consequent need to implement them compatibly with
the limited resources of embedded microprocessors. Secondly,
the necessity to define effective strategies to decode and
extract damage sensitive features directly from compressed
data. While some successful attempts addressing the first issue
have recently emerged [4], [5], comprehensive approaches
which can concurrently optimize both challenges are barely
explored and, hence, represent an open field of research.

The goal of the current work is to fill the gap above by
proposing a novel framework which exploits embedded Sys-
tem Identification (eSysId) as a data compression scheme in
combination with a tiny Machine Learning (TinyML) solution
for the sake of vibration diagnostics at the extreme edge.
Indeed, this TinyML perspective, which is defined as the capa-
bility to run ML/Deep Learning (DL) models at the boundary
between the physical and the digital layers, has profoundly
revolutionized the classical approach to pattern recognition
problems, disclosing novel and unprecedented opportunities
for pervasive artificial intelligence working in a self-contained
and energy efficient manner [6].

A. eSysId in a nutshell

System identification (SysId) refers to a set of signal
processing techniques that postulate a mathematical model
on a given time series, such that the latter can be used
to replicate the underlying system dynamics [7]. Under this
perspective, SysId models are classically used for structural
characterization by estimating a set of defining parameters,
also termed as model parameters, which can encapsulate all
the meaningful system (structural) information.

The key enabling reason for using SysId as a data com-
pressor is that the number of model parameters necessary
for the accurate identification of the system properties is
at least 30x-40x times lower than the total length of the
processed waveform [8]. This unique characteristic allows
for massive compression ratios, which are at least one order
of magnitude higher than those obtainable via standard data
reduction methods [9]. However, despite this very favorable
property, the detrimental aspect of SysId lies in its huge
computational complexity which hampers its straightforward
implementation on edge devices.

Only one work can be found in which successful implemen-
tation of SysId has been achieved by proposing an embedded
system-oriented alternative (eSysId), purposely designed to
work on general-purpose microprocessors [9]. A cost-benefit
analysis has also been performed by the authors, proving
that eSysId running at the extreme edge can improve the
sensor battery life cycle of at least 4x if compared with
standard compression techniques. Leveraging this preliminary
validation, eSysId has then been deployed in [10] as the main
processing core of a novel intelligent sensor unit, proving
high accuracy for the real-time characterization of small-scale
test-beds even in presence of highly quantized (i.e., 8 bit
resolution) vibration data [11]. In the current work, we extend
these previous attempts by showing that the returned model
parameters can be employed for structural assessment directly
in a compressed form, without requiring cumbersome post-
processing steps to be performed at the decoding side.

B. Contribution

The main contributions of the work can be listed as follows:

1) First, we explored for the first time a novel signal pro-
cessing workflow compatible with structural diagnostics
at the extreme edge from compressed data. The solution
is based on eSysId as a data reduction method and a
tiny Convolutional Neural Network (CNN) as anomaly
detector. Results show that this exclusive combination
can overcome the current limitations of state-of-the-art
solutions relying on standard statistical pattern recogni-
tion procedures, while achieving comparable detection
capabilities.

2) Second, we deployed the designed tiny DL model on
different off-the-shelves microcontroller units (MCUs),
assessing the performance in terms of classification
scores, execution time, and energy consumption.

3) Third, we tested the validity of the proposed processing
pipeline on a representative SHM dataset, related to the
condition monitoring of a bridge structure.

The manuscript begins with a detailed description of the
main building blocks of the processing chain, from the data
compression via eSysId (Section II-A) to anomaly detection
built on a tiny CNN (Section II-B). Experimental validation
is presented in Section III, which aims at assessing the
classification performances of the designed TinyML model
running on different microprocessors. Results are discussed in
Section IV, whereas conclusions and future research directions
can be found at the end.

II. FROM EMBEDDED SYSTEM IDENTIFICATION TO
ANOMALY DETECTION

A. eSysId via ARMA model

Autoregressive with Moving Average (ARMA) models be-
long to the class of output-only SysId techniques, meaning that
they can be used to extract meaningful features (i.e., the model

1https://www.tinyml.org/
parameters) when only the measured system response \( r[n] \) is available, and no possibility exists to access the input stimulus \( s[n] \). Noteworthy, this is a crucial requirement for SHM applications, since the exciting forces are the ones provided by operational (passage of vehicles, pedestrians, ground motion, etc.) and/or environmental (wind, etc.) agents. In these cases, \( s[n] \equiv e[n] \sim N(0, \sigma_e^2) \) commonly holds, namely the system input is assumed equal to a zero-mean noise Gaussian term with variance \( \sigma_e^2 \).

The discrete-time mathematical definition of an ARMA time-series model at a generic sample index \( n \) reads as:

\[
r[n] + \sum_{i=1}^{Q} \theta_i r[n-i] = e[n] + \sum_{i=0}^{P-1} \gamma_i e[n-t]
\]  

(1)

In Equation (1), the quantities \( P \) and \( Q \) indicate the number of parameters preserving memory of the past \( P \) input (MA part) and \( Q \) output (AR part) instances, while \( \theta_i \) and \( \gamma_i \) are the feedback and feed–forward system model. The sum \( N_p = P + Q \) equals the total amount of model coefficients to be determined, also known as model order.

According with the standard eSysId-driven vibration diagnostic theory, structural assessment is performed by extracting frequency-related features (e.g., peak spectral values) out of the system power spectrum, the latter being computed via the Fourier transform of Equation (1). Nonetheless, such a conventional approach suffers from two crucial drawbacks:

- the lack of robustness of the adopted feature extraction methods, first among all the peak-picking algorithm, whose actual effectiveness is conditioned upon the unpredictable levels of noise which can easily hinder faint spectral components;
- the impossibility to reconstruct more sophisticated structural parameters (such as complex mode shapes\(^2\)) apart from the most energetic frequencies of vibration, as imposed by the output-only nature of the considered model.

In this work, we overcome the above mentioned limitations by feeding the output of eSysId straight to the input of a tiny ML model, which can be run, in cascade, by the same microprocessor.

B. Anomaly detection via CNN

Convolutional Neural Networks (CNNs) are among the most powerful class of artificial neural networks for DL and proved outstanding performances especially in the field of image data processing [12]. Their architecture is mainly built on two main parts: i) one convolutional block, aiming at creating multiple feature maps by filtering the input values through the cascade of several convolutional layers (eventually followed by pooling and dropout layers necessary for dimension reduction); ii) one classification block, which flattens and combines the produced maps via dense fully-connected layers, such that it is possible to implement regression/classification tasks by specifying the number and the activation function of the final output layer. Compared with alternative architectures, CNN might suffer from poor generalization to novel viewpoints, as well as they imply a gradual undersampling of the input space that can lead to a loss of spatial relationships between data [13]. However, their hierarchical learning mechanism combined a wider receptive field make them a favorable candidate for classification tasks [14] when compared to autoencoders.

For anomaly detection problems, as the one considered in this work, the objective of the CNN is to provide a binary

\(^2\)Mode shapes are defined as the deflection pattern exhibited by a structure at different frequencies. Their spatial-dependent nature makes them insightful parameters for damage localization, beside being used for detection purposes.
classification about the status of the structure, i.e., to predict whether the input data are representative of a healthy (label ‘0’) or defective (label ‘1’) condition. According with this framework, the two-layer one-dimensional CNN in Fig. 1 has been implemented, with input-output quantities henceforth described:

1) Input features: $\beta = [\Theta \ \Gamma \ T]$ is the vector of input features, given by the concatenation of the AR parameters $\Theta = [\theta_1 \ldots \theta_Q]$, the MA parameters $\Gamma = [\gamma_0 \ldots \gamma_P]$, and the temperature value (normalized to a reference temperature of $25^\circ$)\(^3\). Remarkably, it is paramount to underline that, as already demonstrated in the literature \cite{15}, the inclusion of environmental parameters as additional input features of the ML/DL models can significantly increase the scores of the inference process. This avoids burdensome data pre-processing steps necessary to compensate the unavoidable drifts in data due to external phenomena affecting the structure.

2) Output value: the damage status $DS \in [0; 1]$ is returned by the output layer of the network, consisting of one single neuron whose output $o$ is activated by a Sigmoid function followed by a binary thresholding logic

$$DS = \begin{cases} 1, & \text{for } o \geq th \\ 0, & \text{for } o < th \end{cases}$$

(2)

III. EXPERIMENTAL VALIDATION

In this Section, experimental proof of the proposed workflow is presented for the representative SHM scenario of the Z24 bridge dataset.

A. Target structure and related dataset

Built in Switzerland in the early ’60s, the Z24 bridge was at the center of an important highway viaduct between Bern and Zürich. The serviceability of the structure was interrupted in 1999 for modernization; the reason why this structure has gained incredible success within the SHM community is because, before its demolition, it has been artificially damaged via a rigorous experimental protocol, moving from slight damage flaws to very severe disruption actions \cite{16}. Moreover, according with its construction principles, it can be considered as representative of a classical post-tensioned concrete box girder bridge, as the ones currently realized.

A dataset was collected during one year of monitoring, both in healthy and damaged conditions, and publicly released\(^4\). The number of available tests amounts to 5651, each of them comprising both acceleration and environmental readings (such as temperature and humidity) measured at different positions of the bridge: 4923 instances are labelled as healthy, whereas the remaining ones pertain to the structure in defective configurations. For the sake of clarity, it is worth specifying that, being the objective of this work to demonstrate the extreme edge deployment of eSysId jointly with a CNN-driven anomaly detector, the dataset was treated from a sensor-oriented perspective, i.e., only vibration and temperature signals acquired by one sensor (the one installed at the bridge midway) were considered.

B. Offline data processing and inference

The processing pipeline in Fig. 1 was implemented, starting with eSysId model parameter extraction. To this end, the optimal number $N_p$ has firstly been estimated on a reduced batch of acceleration signals, by adoption of the Akaike Information Criterion \cite{17}: the latter represents a well-known statistical metric used to find $N_p$ as that value minimizing the error between the actually measured and the numerically predicted system response in Equation (1). A model order $(P, Q) = (20, 20)$ has been judged sufficient for the class of processed time series, which consist of 5000 samples acquired at 50 Hz, yielding to an achievable compression ratio higher than 125x.

As such, a vector $\beta$ of 41 elements (20 AR parameters, 20 MA parameters and 1 normalized temperature value) has been obtained and provided as input to the tiny CNN model, which has been trained and validated via Adam optimizer (learning rate = 0.001) with the following configuration: train/test split of 70%-30%, loss function coincident with the mean square error, batch size equal to 32. Finally, a threshold $th = 0.5$ has been set for binarization purposes at the end of the output layer.

C. TinyML deployment: from quantization to on-edge inference

The embodiment of ML/DL models on resource-constrained devices requires additional steps in order to fit the computational and memory constraints of the target microprocessor. First of all, the model has to be converted into a tiny fashion by means of ad-hoc quantization strategies, whose effect is to cast the model weights and biases (and the input/output data) to a convenient data type and bit-width, compatible with the digital signal processing functionalities of the target board. Since this procedure might introduce performance degradation, the selection of the optimal quantization technique assumes pivotal importance and represents a pivotal point of the TinyML deployment process \cite{18}. As such, practitioners and researchers have started to explore different solutions, including pre- or post-training quantization with float/integer data format. In this work, a quantization aware (QA) solution with float32 input/output has been preferred, thanks to the fact that it can minimize the potential loss by emulating inference-time quantization directly during the training stage \cite{18}. QA has the additional advantage of totally preserving the accuracy of the model after conversion, but typically leads to longer training time \cite{19} and could imply a worst overall performance with respect to post-training quantization.

After conversion, the generated model resulted in a total complexity of 343381 multiply and accumulate operations, with a memory occupancy of 4.85 KBytes of RAM and

\(^3\)From the above definition, it is immediate to observe that the length of $\beta$ amounts to $N_p + 1$.  
\(^4\)https://bwk.kuleuven.be/bwm/z24
50.75 KBytes of flash, hence being deployable on the majority of cut-of-the-shelves MCUs available on the market.

To this end, the Application Programming Interface made available by the TensorFlow (TF) and TensorFlow Lite libraries were exploited in this work. Besides, the actual deployment process was fulfilled by resorting to dedicated software and expansion packages, such as the STM32 X-CUBE-AI, which offer native support for the generation and integration of various pre-trained models.

1) Prototyping boards: Three different evaluation boards were considered for prototyping purposes: the Arduino Nano 33 BLE Sense board [20], and two STMicroelectronics Nucleo boards, namely the STM32L496ZG-P [21] and the STM32L522ZE-Q [22] board. Their relevant characteristics in terms of embedded MCU, storage capabilities, clock speed and power consumption (in run mode) are measured for the Arduino board. Conversely, the obtained metrics compare favorably with the TF counterpart for all the considered classification targets, with minimal deviation in between them (below 0.3%).

Additionally, the time taken to run a single inference has been measured to verify the suitability of the proposed workflow in view of real-time diagnostic functionalities. The minimum inference time has been registered for the STM32L5 board, with an average processing time of 30 ms/inf, followed, in order, by the Arduino and the STM32L4 board: it is worth noting that, besides these differences which are largely dominated by the clock speed and the computing architecture, these quantities are absolutely compatible with the typical duty-cycles of the target application context, where monitoring is usually performed on a hourly basis due to the long inertia of the aging defects.

Finally, a first-order quantitative analysis about the impact of the overall processing flow on the energy budget of a candidate sensor node has been included: assuming that the edge sensor is powered by a battery with a capacity of 3600 mAh and inspection (summing the contribution of the eSysId as reported in [9] and the current tiny CNN) is performed 24 times/day, the resulting system can work autonomously for up to 3 years depending on the select MCU. Consequently, the proposed strategy appears as a promising candidate for sensor-near vibration-based SHM, opening new solutions for long-lasting monitoring systems.

IV. RESULTS

The effectiveness of eSysId has extensively been characterized in previous works [9]–[11]. As such, emphasis has deliberately been posed to the subsequent TinyML-driven anomaly detection block. Results for the tiny CNN implementation on the three targets are reported in Table I (TinyML performance section), also including the classification scores pertaining to the offline implementation in the TF framework. More in detail, four standard anomaly detection metrics have been computed [23]:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

\[
F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

in which \(TP, TN, FP, FN\) represent the number of true/false positives/negatives.

As can be seen in Table I, the performances on MCUs are very close to the reference ones: the maximum accuracy and precision loss amounts to 1.77% and 1.32%, respectively, and are measured for the Arduino board. Conversely, both the STM32 realizations are always in good agreement with the TF counterpart for all the considered classification scores, with minimal deviation in between them (below 0.3%). Furthermore, the obtained metrics compare favorably with respect to previous TinyML attempts investigated in the field on the same dataset [24]. The fundamental advantage of our implementation is that we can can bypass the intense and bulky data processing workload resorted by the authors in [24] for the extraction of damage sensitive features according with conventional vibration analysis techniques, which are far from being portable at the extreme edge.

V. CONCLUSIONS AND FUTURE WORKS

This work aimed at presenting a novel approach for vibration-based structural diagnostics at the extreme edge. The solution is peculiar in that it combines, for the first time, the utmost benefits of eSysId as an unconventional data compression scheme with the computational and energy advantages of a light CNN model in solving anomaly detection problems. The envisioned framework has been prototyped on three general-purpose microprocessors, working on the representative dataset of the Z24 bridge in Switzerland. Results demonstrate that the quantized models running on the target platform can achieve classification scores always above 90% with a negligible loss in performances if compared with offline implementations.

Future improvements will move in two main directions. From one side, the entire pipeline, from data acquisition to damage detection, will be tested in real-time for the diagnosis of large-scale test-beds. On the other hand, alternative tiny ML/DL models and/or knowledge distillation strategies will be explored to further shrink the algorithmic complexity and, in turn, the energy expenditure.

REFERENCES


### TABLE I
**Tiny CNN for eSysId-driven vibration diagnostics: main features of the adopted prototyping boards together with classification scores and performance parameters.**

<table>
<thead>
<tr>
<th>Features</th>
<th>Arduino Nano 33 BLE Sense</th>
<th>Nucleo STM32L496ZG-P</th>
<th>Nucleo STM32L522ZE-Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU</td>
<td>Nordic nRF52840</td>
<td>ARM® Cortex®-M4</td>
<td>ARM® Cortex®-M33</td>
</tr>
<tr>
<td>RAM</td>
<td>256 KBytes</td>
<td>320 KBytes</td>
<td>256 KBytes</td>
</tr>
<tr>
<td>flash</td>
<td>1 MBytes</td>
<td>1 MBytes</td>
<td>512 KBytes</td>
</tr>
<tr>
<td>Maximum clock speed</td>
<td>64 MHz</td>
<td>80 MHz</td>
<td>110 MHz</td>
</tr>
<tr>
<td>Absorbed current (run mode)</td>
<td>6.30 mA @3.3 V</td>
<td>7.28 mA @3.3 V</td>
<td>6.82 mA @3.3 V</td>
</tr>
<tr>
<td>Offline performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>91.95</td>
<td>91.10</td>
<td>90.81</td>
</tr>
<tr>
<td>Precision</td>
<td>94.05</td>
<td>92.81</td>
<td>92.55</td>
</tr>
<tr>
<td>Recall</td>
<td>96.83</td>
<td>96.60</td>
<td>97.72</td>
</tr>
<tr>
<td>F1</td>
<td>95.42</td>
<td>94.64</td>
<td>95.04</td>
</tr>
<tr>
<td>Execution time</td>
<td>35 ms/inf</td>
<td>44 ms/inf</td>
<td>30 ms/inf</td>
</tr>
<tr>
<td>Energy consumption/inf</td>
<td>127.60 mJ/inf</td>
<td>1067.60 mJ/inf</td>
<td>675.18 mJ/inf</td>
</tr>
<tr>
<td>Sensor life cycle</td>
<td>2.34 years</td>
<td>2.42 years</td>
<td>3.19 years</td>
</tr>
</tbody>
</table>


