On the use of Echo State Networks for Anomaly Detection in Industrial IoT Systems through On-Device Training

Fabrizio De Vita, Giorgio Nocera, Dario Bruneo

Department of Engineering, University of Messina, Italy
CINI Italian Lab on Artificial Intelligence and Intelligent Systems (AIIS)
fdevita@unime.it, giorgio.nocera@studenti.unime.it, dbruneo@unime.it

Abstract

The advent of the Industrial Internet of Things (IIoT) technology has significantly changed the interaction with the world and systems in general. Indeed, the use of this technology had a strong impact in the industrial context, leading to the birth of a new paradigm usually called Industry 4.0. In such a context, hundreds of devices with sensing/actuating capabilities communicate between them and with the surrounding environment (for this reason called Cyber Physical Systems), actively cooperating to perform one or more tasks. In particular, anomaly detection became a crucial topic in the industrial context to reduce unwanted (or unplanned) maintenance and avoid potentially dangerous conditions that can compromise a system operability. Machine learning is a very important component for bringing intelligence to Cyber Physical Systems (CPS) representing one of the main building blocks for the realization of a new category of systems called Intelligent Cyber Physical Systems (ICPS). In this sense, the majority of the anomaly detection techniques proposed in the literature today involves the Artificial Intelligence (AI) that provides the tools for the diagnosis of a system "health" state by analyzing sets of sensor of various nature. In this work, we present an anomaly detection algorithm based on Echo State Networks (ESNs) for the analysis of sensor data. Using compression and quatization approaches we deployed it on a microcontroller of the STM32 family performing an on-device training and enabling a real time monitoring of an industrial plant.

1 Introduction

During these years, we observed a market growth of Internet of Things (IoT) applications in smart industry. The use of this technology in the industrial sector has significantly changed the way we interact with these system, leading to what we call Industrial Internet of Things (IIoT) and representing the core element of the new Industry 4.0 paradigm [Vallati *et al.*, 2019; Sisinni *et al.*, 2018]. In such a context, a timely and effective detection of conditions (or events) which should not be considered normal (i.e., anomalous) can be crucial to avoid potentially dangerous situations that can lead to the complete industrial system breakdown.

Until recently, Cloud computing has been a central component in the realization of anomaly detection frameworks by providing the infrastructure, the storage, and the high computing power necessary to run complex algorithms. On the other hand, if we consider an industrial scenario application posing requirements in terms of low latency response times, data privacy, and stable Internet connections, the use of this paradigm becomes ineffective [Parikh *et al.*, 2019].

For all these reasons, modern solutions involve the Edge computing to address the above mentioned problems by shifting the computation "close" to where the data is generated. However, the hardware constraints of Edge devices pose significant limitations on the jobs they can execute. Artificial Intelligence(AI) is another very important player for the implementation of smart systems such as Intelligent Cyber Physical Systems (ICPS) that can exploit their sensing/actuating capabilities to perform several tasks. Through machine learning techniques, we are able to introduce a new "cognition" layer that enables a system to provide a "reasoned" support to the human being. In this sense, if the one hand AI is fundamental to bring the intelligence to services and applications, on the other, the limited resources of Edge devices make the execution of onerous algorithms (e.g., neural networks models) very challenging [De Vita *et al.*, 2021]. Evidently, this problem becomes even more complex considering the large number of variables and non-linear relationships that can emerge when working with industrial plants.

Approaches like compression and quantization allow to mitigate this problem by reducing the complexity of a machine learning model such that it can fit the hardware constraints of an Edge device, while keeping its performance. Compression is a technique that aims to reduce the number of weights by "condensating" them into a limited set of clusters, while maintaining the original number precision. On the other hand, the quantization allows to keep the same amount of weights of the original model, but sensibly reduces their memory occupation by projecting them on a discrete interval space (typically with a 8 bit resolution).

In this work, we present an anomaly detection algorithm based on Echo State Networks (ESNs) [Jaeger, 2001] for the analysis of multivariate time series data. Exploiting compres-

sion and quantization techniques, we were able to perform an on-device training of the algorithm directly on a microcontroller of the STM32 family, thus enabling a real time monitoring of an industrial plant. Experimental results show that the proposed application is able to correctly detect the occurrence of anomalies in the system and demonstrate the feasibility of the proposed approach for the realization of smart Edge monitoring platforms.

2 Echo State Networks

Echo State Netowrks (ESNs) are a family of neural networks belonging to the Reservoir Computing (RC) framework characterized by very sparsely connected hidden layers and particularly suitable for the analysis of time series data [Jaeger, 2001]. Figure 1 shows a typical ESN architecture where the dashed and solid arrows represent respectively the trainable and the not-trainable weights, while the gray lines indicate valid but not required links.

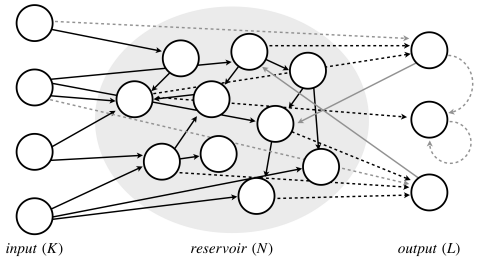

Figura 1: ESN architecture.

In such a network, the reservoir weights are fixed after a random initialization and the only weights modified during the training stage are the ones that connect the reservoir to the output. Such a feature allows to decrease the model complexity because of the lower number of trainable weights and to speed-up the training phase which can be achieved solving a linear regression problem [Jaeger, 2001]. Given a network with K input units, N reservoir units, and L output units, an ESN is described by the following matrices: a $N \times K$ input matrix W_{in}, a $N \times N$ reservoir matrix W_{res}, a $L \times (K + N + L)$ output matrix W_{out}, and a $N \times L$ back-projection matrix W_{back}.

More in detail, the reservoir state update is governed by the following equation:

$$x(t+1) = f(W_{in} \cdot u(t+1) + W_{res} \cdot x(t) + W_{back} \cdot y(t)), \quad (1)$$

where, $u(t+1)$ is the new input, $x(t)$ is the previously computed state, $y(t)$ is the previous output, and $f(\cdot)$ is the activation function (tipically a *sigmoid* or a *tanh* function).

With respect to the output, it is computed through the following equation:

$$y(n+1) = g(W_{out} \cdot [u(t+1), x(t+1), y(n)], \quad (2)$$

where $g(\cdot)$ is the activation function (tipically a sigmoid or the identity), $u(t+1)$ is the new input, $x(t+1)$ is the new computed state and $y(t)$ is the previous output.

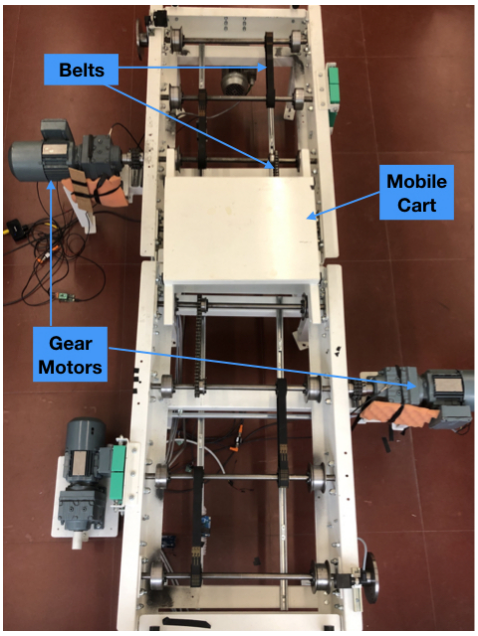

Figura 2: Industrial assembly plant.

3 Case study

Thanks to our collaborations with companies expert in the automotive sector, we are currently applying our technique on a real case study represented by a scale replica of an assembly plant (shown in Figure 2) used for the transportation of car pieces in automobile plants. The system is equipped with two gear motors, and six belts (four made of rubber, and two made of steel) to transport the mobile cart. An important feature of the plant is the possibility to inject different types of faults (mainly mechanical) such as: the introduction of external vibrations, change the belt tension, increase the friction of the gears, and emulate the break of the cart proximity switch. Such a feature has been fundamental for our purposes, since it allowed us to have a complete understanding of its dynamics even when subject to a mechanical stress. The system has been also instrumented using several type of sensors (e.g., current, vibration, temperature, noise, and distance) in order to perform the telemetry from both the mechanical and electrical point of views.

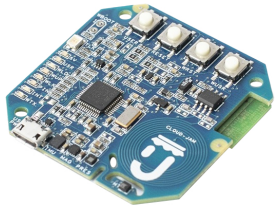

Figura 3: Cloud-JAM board.

In our experimentation, we adopted a STMicroelectronics smart board called Cloud-JAM (shown in Figure 3) where we deployed our anomaly detection algorithm based on ESNs. Using approaches like K-Means weights compression and 8-

bit integer quatization, we were able to strongly reduce the memory footprint of our model and make it suitable to fit the hardware constraints of the smart board. This gave us the opportunity to perform both the training and inference processes on the STM32 device microcontroller. Hence, our model is able to run a real-time anomaly detection algorithm which makes it suitable to be used in industrial application scenarios where normal and anomalous patterns are dynamic and evolve over time requiring a constant analysis and monitoring.

4 Preliminary results

In this section, we present the results derived from testing the performance of our model. The algorithm has been first trained on a "working" environment to make it learn the features and patterns describing this condition. After this phase, we started a real-time injection of several faults to test its ability in detecting the anomalies on a dynamic setting. Figure 4 depicts the confusion matrix computed on the test set from which we extracted some indicators such as: $precision$, $recall$, and $F1\text{-}score$.

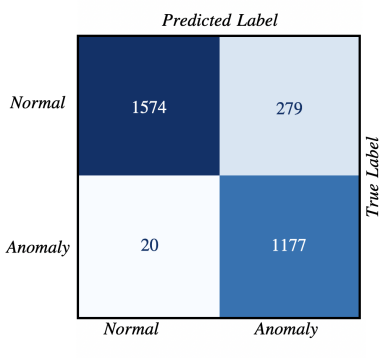

Figura 4: Anomaly detection confusion matrix.

Our model resulted in high values for each performance metric reaching a precision of 0.807, a recall of 0.983 and an overall $F1\text{-}score$ of 0.887. The obtained results (albeit preliminary) demonstrate the feasibility of our technique and encourage us to investigate new approaches in order to further improve the performance of our algorithm.

Riferimenti bibliografici

[De Vita *et al.*, 2021] Fabrizio De Vita, Giorgio Nocera, Dario Bruneo, Valeria Tomaselli, Davide Giacalone, e Sajal K. Das. Porting deep neural networks on the edge via dynamic k-means compression: A case study of plant disease detection. *Pervasive and Mobile Computing*, 75:101437, 2021.

[Jaeger, 2001] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[Parikh *et al.*, 2019] Shalin Parikh, Dharmin Dave, Reema Patel, e Nishant Doshi. Security and privacy issues in cloud, fog and edge computing. *Procedia Computer Science*, 160:734–739, 2019.

[Sisinni *et al.*, 2018] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, e M. Gidlund. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Transactions on Industrial Informatics*, 14(11):4724–4734, Nov 2018.

[Vallati *et al.*, 2019] C. Vallati, S. Brienza, G. Anastasi, e S. K. Das. Improving Network Formation in 6TiSCH Networks. *IEEE Transactions on Mobile Computing*, 18(1):98–110, Jan 2019.