Detection of Component Degradation: A Study on Autoencoder-based Approaches

Dario Guidotti
DUMAS
University of Sassari
Sassari, Italy
dguidotti@uniss.it

Laura Pandolfo
DUMAS
University of Sassari
Sassari, Italy
lpandolfo@uniss.it

Luca Pulina
DUMAS
University of Sassari
Sassari, Italy
lpulina@uniss.it

***Abstract*—In the realm of predictive maintenance, the incorporation of artificial intelligence (AI) methods has revolutionized the field by empowering businesses to actively monitor and preemptively address equipment malfunctions. Detecting anomalies plays a crucial role in predictive maintenance as it serves as an early indicator of potential faults or failures. This paper introduces initial findings from the use of autoencoders and their associated vector reconstruction error within the context of the IMOCO4.E project.**

***Index Terms*—Predictive Maintenance, Anomaly Detection, Neural Networks**

I. Introduction

Predictive maintenance using AI has transformed industrial systems by proactively identifying and addressing potential failures. Anomaly detection [1] plays a crucial role in this approach, monitoring equipment behavior for deviations from normal patterns. Autoencoders [2], a type of neural network, have emerged as a promising method for anomaly detection. The IMOCO4.E project [3] aims to enhance mechatronic systems' intelligence and adaptability by integrating novel sensory information, model-based approaches, AI, ML, and industrial IoT principles. In this work, we focus on a cutting machine in the food and beverage industries and aim to build an anomaly detection system for blade degradation. The paper presents our solution, experimental results, and future research plans.

II. Case Study

In this particular investigation, we center our attention on the One Year Industrial Component Degradation Dataset [4]. This dataset captures a comprehensive set of measurements acquired over the course of one year from an OCME Vega shrink-wrapper, a machine employed in the packaging industry. The dataset consists of sampling sessions with a duration of 8 seconds and a time resolution of 4 milliseconds. Each session encompasses 2048 samples, and the number of sessions varies across different months, yielding a rich dataset comprising a total of 1,062,912 samples. Various sensors installed on the machine contribute to the dataset, with primary emphasis placed on monitoring the performance of the cutting blade component, responsible for slicing the packaging plastic film. The specific measurements examined in this study involve motor torque, blade and film position, blade and film speed, the time lag between expected and actual positions of the blade and film, as well as a performance evaluation metric.

The fundamental objective of this case study revolves around developing a detection system that can effectively recognize instances where the cutting blade component experiences degradation, indicating the need for replacement. By thoroughly analyzing the collected data and employing suitable training techniques, such as autoencoders, our aim is to construct an anomaly detection model that facilitates timely maintenance and replacement of the cutting blade component, ensuring the continuous optimal performance of the packaging process.

III. Methodology

In this preliminary analysis, we evaluate two different architectures for our autoencoders. Both architectures have three hidden linear layers followed by ReLU activation functions. The output layer present an identity activation function. The first autoencoder ($A1$) has 32, 8, and 32 hidden neurons in the first, second, and third hidden layers, respectively, whereas the second autoencoder ($A2$) has 64, 16, and 64 hidden neurons in the same layers. As can be seen, our models presents an expansion of the output space in the first and third hidden layers and a contraction in the second hidden layer. This structures can be interpreted as an encoder and a decoder, which work together to learn a compact representation of input data: the encoder maps the input data into a lower-dimensional latent space representation, capturing the most salient features and patterns of the input data and effectively reducing its dimensionality; The decoder then reconstructs the original input data from the encoded representation. During training, autoencoders aim to minimize the reconstruction error between the input and output data. By optimizing this reconstruction loss, the autoencoder learns to capture the underlying structure of the data, enabling it to generate accurate reconstructions.

To quantify the magnitude of an anomaly, we utilize the *vector reconstruction error* (VRE), following an approach similar to [5]. The VRE measures the discrepancy between the reconstructed output and the corresponding input data. By assuming that the autoencoder has been properly trained, this measurement enables the recognition of anomaly presence and

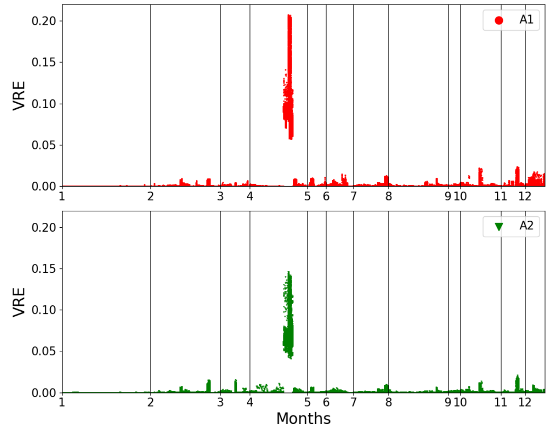

Fig. 1. Graphical representation of the results of our experimental evaluation. On the x-axis are reported the input samples of interest divided in the 12 months of measurements, whereas on the y-axis the VRE of our models is reported.

magnitude. The underlying principle is that if the autoencoder fails to accurately reconstruct the input data, then such data is anomalous in some way. In our experimental evaluation, we employ the mean square error (MSE) between the output and the corresponding target as the VRE.

IV. Training Process

We employed pyNeVer [6], a neural network management, training, and verification tool, to train the networks of interest. pyNeVer utilizes PyTorch [7] as a backend and provides a user-friendly custom training loop.

For training, we utilized the Adam optimizer with a learning rate of 0.001 and the mean square error as the loss function. The networks were trained for 50 epochs, with a batch size of 512 for training and 128 for validation and testing. The validation set constituted 30% of the training data, while 20% of the dataset was reserved for testing. These learning parameters remained consistent across all trained networks.

It is worth noting that, for anomaly detection tasks, the autoencoders must be trained on a subset of the dataset devoid of anomalies. Therefore, in our training and testing procedures, we only considered the first 200,000 samples of the dataset. This choice is based on the reasonable assumption that significant blade degradation would not occur within the initial period of operations.

V. Results

The experiments were conducted on a MacBook Air laptop equipped with an Apple M2 CPU and 24 GB of RAM. The operating system used was macOS Ventura 13.4. To train the autoencoders, we employed MPS. The code necessary to replicate our experiments can be found in our repository [8].

Both our autoencoders reached reasonable performance during the training phase. Specifically, the mean square errors (MSEs) computed on the test set were $5.72 \cdot 10^{-7}$ and $3.59 \cdot 10^{-6}$ for $A1$ and $A2$ respectively. Notably, increasing the number of neurons in the hidden layers did not lead to a significant enhancement in network accuracy.

In Figure 1, the scatter plot representing the vector reconstruction error for our autoencoders is presented. Remarkably, an anomaly is detected in the data during the 4th month. Even the smallest MSE value computed during the anomaly is significantly higher than the maximum VRE observed in the non-anomalous data points. It is important to note that while $A1$ exhibits the greatest proficiency in identifying the anomaly, both the autoencoders demonstrate the ability to correctly detect it. As observed previously in terms of network accuracy, there appears to be no significant correlation between the size of the autoencoders and their performance in anomaly detection.

VI. Future Work

Our current focus involves expanding our experimental evaluation and further investigating the other minor anomalies observable during the cutting blade's operation. Understanding the underlying causes of these anomalies is an intriguing avenue for exploration [9]. We plan to explore additional network architectures to assess their impact on autoencoder performance. Additionally, we are interested in utilizing formal verification techniques to enhance the reliability of our autoencoders [10], [11].

Acknowledgement

This work was supported by the H2020 ECSEL JU grant agreement No. 101007311 "Intelligent Motion Control under Industry4.E" (IMOCO4.E) project. The support is gratefully acknowledged.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[2] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Unsupervised and Transfer Learning - Workshop held at ICML 2011*, ser. JMLR Proceedings, vol. 27. JMLR.org, 2012, pp. 37–50.

[3] M. Cech, A. Beltman, and K. Ozols, "Digital twins and AI in smart motion control applications," in *Proceedings of ETFA 2022*. IEEE, 2022, pp. 1–7.

[4] "One year industrial component degradation dataset," https://www.kaggle.com/datasets/inIT-OWL/one-year-industrial-component-degradation, accessed: 2023-07-14.

[5] H. Torabi, S. L. Mirtaheri, and S. Greco, "Practical autoencoder based anomaly detection by using vector reconstruction error," *Cybersecur.*, vol. 6, no. 1, p. 1, 2023.

[6] D. Guidotti, L. Pulina, and A. Tacchella, "pynever: A framework for learning and verification of neural networks," in *Proceedings of ATVA 2021*, ser. Lecture Notes in Computer Science, vol. 12971. Springer, 2021, pp. 357–363.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of NIPS 2019*, 2019, pp. 8024–8035.

[8] "imoco4e-eScience2023," https://github.com/darioguidotti/imoco4e-eScience2023, accessed: 2023-07-15.

[9] D. Guidotti, R. Masiero, L. Pandolfo, and L. Pulina, "Vector reconstruction error for anomaly detection: Preliminary results in the imoco4.e project," in *to appear*, 2023.

[10] D. Guidotti, L. Pandolfo, and L. Pulina, "Verification of nns in the imoco4.e project: Preliminary results," in *to appear*, 2023.

[11] ——, "Leveraging satisfiability modulo theory solvers for verification of neural networks in predictive maintenance applications," *Information*, vol. 14, no. 7, p. 397, 2023.