Anomaly Recognition with Trustworthy Neural Networks: a Case Study in Elevator Control

Dario Guidotti[1,*], Laura Pandolfo[1] and Luca Pulina[1]

[1]University of Sassari, Piazza Università 21, Sassari, 07100, Italy

Abstract

In the realm of contemporary industrial control systems, the necessity for robust anomaly detection and classification is of critical importance. This paper presents an application of neural network technology in a real-world industrial scenario focused on elevator control. We employ two fully-connected neural networks to accomplish both anomaly detection and classification. The first neural network is dedicated to identifying types of anomalies, while the second predicts their magnitudes. Additionally, we integrate formal verification to certify the local robustness of these networks. Our findings not only showcase the practical efficacy of our methodology but also emphasise the crucial role of small neural networks in effectively addressing challenges within industrial settings.

Keywords

Anomaly Detection, Neural Networks, Formal Verification, Trustworthy AI

1. Introduction

In industrial automation and control systems, the smooth operation of critical machinery is essential across various sectors. Elevator control systems are crucial in settings like commercial buildings, residential complexes, and industrial facilities, efficiently transporting people and goods under demanding conditions. As elevator control systems' complexity has grown to meet modern infrastructure needs, so have challenges in maintenance and operation. Detecting and classifying anomalies in these systems is critical for accident prevention, minimising downtime, and reducing maintenance costs. These anomalies range from mechanical faults to sensor errors, requiring resolution for safe and reliable elevator operation. Traditionally, rule-based methods and hand-crafted algorithms handled anomaly detection in elevator control systems. While effective to some extent, they struggle with the diverse and dynamic nature of real-world anomalies. Additionally, they may lack the ability to quantify the severity of detected anomalies, limiting the prioritisation of maintenance efforts. The rise of artificial intelligence (AI), especially neural networks (NNs), has transformed anomaly detection and classification [2, 3]. These technologies, capable of learning intricate patterns from data, offer a promising avenue to enhance the reliability and effectiveness of anomaly detection in elevator control systems.

This paper introduces a real-world application of NNs in the domain of elevator control systems, focusing on achieving three primary objectives:

- **Anomaly Detection and Classification**: This work propose to employ two fully-connected NNs to identify and categorise anomalies in elevator control systems. The first NN focuses on discerning the type or category of the identified anomaly, providing valuable insights into its characteristics. The second NN is specifically tailored for regression, forecasting the magnitude or severity of the anomaly, thus providing a measure of its impact.
- **Formal Verification for Robustness**: Recognising the safety-critical nature of elevator control systems, special emphasis is placed on certifying network robustness. To guarantee the reliability

AI4CC-IPS-RCRA-SPIRIT 2024: International Workshop on Artificial Intelligence for Climate Change, Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy. November 25-28th, 2024, Bolzano, Italy [1].

[*]Corresponding author.

dguidotti@uniss.it (D. Guidotti); lpandolfo@uniss.it (L. Pandolfo); lpulina@uniss.it (L. Pulina)
0000-0001-8284-5266 (D. Guidotti); 0000-0002-5785-5638 (L. Pandolfo); 0000-0003-0258-3222 (L. Pulina)

of the networks, formal verification techniques [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31] can be applied to validate their local robustness. This verification process ensures that the networks' behaviour remains stable within a predefined input space, even when subjected to adversarial perturbations.
- **Real-life Application in Elevator Control**: The viability and efficacy of this methodology are showcased through a practical examination in elevator control. Through the application of NN technology in this authentic industrial setting, our objective is to highlight the significance of our approach in tackling the challenges inherent in contemporary industrial automation.

Our aim is to contribute to the continual improvement of the safety, reliability, and efficiency of elevator control systems. Through the utilisation of NNs and formal verification techniques, we present a sturdy framework for detecting and classifying anomalies, readily applicable in industrial settings. This guarantees the sustained safe and reliable operation of elevators across diverse and dynamic environments. Moreover, through our experimental evaluation, we demonstrate that even simple and compact network architectures remain pertinent in real-world industrial scenarios, thereby affirming the effectiveness of existing verification techniques within this domain.

The rest of the paper is structured as follows. In Section 2 we introduce some basic concepts and definitions, while in Section 3 we provide an overview of the related works. In Section 4 we present the use case of interest and the related dataset and in Section 5 we present our models and the specific of the properties considered in the experimental evaluation. In Section 6 we present the results of our experimental evaluation. Finally, in Section 7 we present our conclusions and some of our future research plans.

2. Background

2.1. Neural Networks

A NN constitutes an intricate structure of computing units, commonly referred to as "neurons". In the context of feed-forward NNs, these neurons are systematically organised into sequential layers. Each neuron within a layer exclusively connects to those in the subsequent layer, forming a sequential flow of information. In a computational sense, a feed-forward NN with p layers can be defined as a function $\nu : \mathbb{R}^n \to \mathbb{R}^m$, where $\nu(x)$ is calculated iteratively as $\nu(x) = f_p(f_{p-1}(...f_1(x)...))$. Here, $f_1 : \mathbb{R}^n \to \mathbb{R}^{n_1}, \ldots, f_p : \mathbb{R}^{n_{p-1}} \to \mathbb{R}^m$ represent the functions corresponding to the layers of the network. Notably, the first layer, f_1, is known as the input layer, the final layer, f_p, serves as the output layer, while any intermediate layers are commonly referred to as hidden layers. We primarily focus on two types of layers:

- *Affine Layers*: These layers implement linear transformations, denoted as $f : \mathbb{R}^n \to \mathbb{R}^m$, and are defined by the equation $f(x) = Wx + b$. Here, W is a weight matrix in $\mathbb{R}^{m \times n}$, and b is a bias vector in $\mathbb{R}^m$. Affine layers are fundamental for processing and transforming input data.
- *ReLU Layers*: Introducing non-linearity to the NN, ReLU (Rectified Linear Unit) layers are characterised by the function $f : \mathbb{R}^n \to \mathbb{R}^n$, defined as $f(x) = max(0, x)$. This element-wise operation enhances the network's capability to model complex relationships within the data.

In this study, NNs have a dual role, addressing both *regression* and *classification* tasks, each with distinct objectives. For regression, the NN's aim is to approximate an unknown functional mapping from $\nu : \mathbb{R}^n \to \mathbb{R}^m$. This involves predicting continuous output values based on input data. In our context, this translates to the task of estimating anomaly magnitudes using sensor measurements. In the case of classification, the focus shifts to categorising input data into predefined classes or categories. In this work, classification involves the task of identifying anomaly categories based on sensor data.

2.2. Formal Verification

The primary objective of formal verification is to prove that a NN's behaviour aligns with specific properties. In particular, our focus is on input-output specifications. These specifications assume that given a precondition on the input, a corresponding postcondition must hold for the output. For a practical example of this kind of property we refer to Subsection 5.2. In recent years, considerable progress has been made in methodologies and tools aimed at addressing verification tasks, as detailed in [32, 33]. We present a succinct overview of state-of-the-art verification techniques based on the categorisation proposed in [32]. This classification is derived from the type of guarantees these techniques offer. *Deterministic guarantees* provide precise information about whether a property is satisfied. Methodologies in this category transform the verification problem into a set of constraints, which are then solved using appropriate solvers. Noteworthy examples include methodologies presented in [34, 6], leveraging SMT solvers, and those in [11, 12, 35], using Mixed Integer Linear Programming (MILP) solvers. Some of the latest methodologies enhance these solvers with techniques like branch-and-bound (BnB). *One-sided guarantees* offer either lower or upper bounds for a variable, serving as sufficient conditions for a property to hold. Approaches in this category can effectively verify larger models and are less susceptible to numerical instability, a concern in methods relying on floating-point arithmetic. These methodologies make use of technologies such as abstract interpretation [22, 9, 10, 36], convex optimisation [37], interval analysis [5], symbolic interval propagation [38], and linear approximation [39, 40]. Finally, *converging guarantees* offer guarantees converging lower and upper bounds and are particularly effective for verifying large models. They employ technologies such as layer-by-layer refinement and analysis [41], reduction to a two-player turn-based game [42, 43], and global optimisation [44, 45].

3. Related Works

In the quest for effective anomaly recognition systems, numerous methodologies and techniques have been investigated and proposed. Considering the vastness of the literature corpus on this subject and the extent of our work, we hereby provide an overview of some pertinent studies and delineate how our work distinguishes itself from them.

In a recent work [46], a novel neural network architecture integrating a variational autoencoder and a generative adversarial network is proposed for detecting anomalies in software runtime execution traces. In [47], the authors present an anomaly detection framework for spacecraft multivariate time-series data, employing temporal convolutional networks. Meanwhile, [48] proposes a recurrent neural network architecture for identifying anomalies in electrocardiograms. Additionally, [49] introduces a novel self-supervised attentive generative adversarial network for discerning unexpected events in videos. Furthermore, [50] and [51] propose two distinct methodologies for anomaly detection in Industrial Control Systems. The former combines a one-dimensional convolutional neural network with a bidirectional long short-term memory network, while the latter utilises a lightweight long short-term memory variational autoencoder to detect cyber-attacks.

This brief overview illustrates the successful application of various neural network architectures for anomaly detection across diverse domains. However, all these models tend to be complex, and their reliability cannot be guaranteed using current state-of-the-art methodologies for neural network verification. In contrast, our paper demonstrates that even simple, straightforward architectures can achieve satisfactory performance levels when applied to specific real-world applications, such as elevator control. Furthermore, we establish the feasibility of certifying the local robustness of such models using state-of-the-art neural network verification tools.

4. Use Case: Elevator Control

IMOCO4.E [52, 53] is a Key Digital Technologies Joint Undertaking (KTD JU) project that commenced in September 2021. The project involves the collaboration of 46 partners spanning 13 countries and is dedicated to advancing mechatronic systems by augmenting their intelligence and adaptability. This objective will be realised through the integration of state-of-the-art technologies, encompassing innovative sensor data, model-based methods, AI, machine learning (ML), and principles of the industrial Internet of Things (IoT). By leveraging these advanced technologies, the project aims to catalyse the evolution of European manufacturing towards Industry 4.0, facilitating the perception and management of intricate machinery and robotics. The project's focal point is the delivery of a reference architecture, subject to testing and validation across diverse industrial domains through various use cases and pilot projects. These applications span sectors such as packaging, industrial robotics, healthcare, and semiconductor manufacturing. In our research, we concentrate on a specific use case related to the application of NNs in the realm of elevator predictive maintenance. Elevators, being intricate machines, necessitate regular maintenance to ensure their safe and reliable operation. The control of elevator movement in tall buildings poses substantial challenges due to non-linearities, uncertainties, and the dynamics of time-varying systems. By harnessing NNs for predictive maintenance, potential issues can be identified before they escalate into breakdowns or pose safety risks. This proactive approach not only minimises downtime but also amplifies the overall performance and safety of elevators.

4.1. Dataset

In this specific instance, we employed data generated from simulating an authentic elevator system. Such simulation was developed and managed by Siemens, which is one of the company involved in the project. This dataset consists of sensor measurements gathered during various journeys of the simulated elevator. During these simulations, three distinct types of anomalies were deliberately introduced, each varying in magnitude, by adjusting the simulation parameters. Specifically, the degradation of the elevator's pulley, its sliding motion, and damaged bearings were simulated as independent factors, in addition to modelling the elevator's standard behaviour. The data collected from the simulation underwent preprocessing by domain experts at WEG, another partner of the project, to extract 187 pertinent features for each elevator journey between two different floors. As a result, the resulting dataset presents a single data point with 187 features for each journey. The target measurements include the classification of the anomaly type (either ageing, bearing or sliding) and the computation of the *performance index* (PI) by WEG. This performance index indicates how much the elevator's behaviour deviates from its nominal state, with higher values signifying greater deviation. It is noteworthy that these target measurements are treated separately: one model is trained for anomaly classification, while another is designed to compute the performance index. It is essential to highlight that all numeric data, including the PI, underwent normalization to fall within the range of 0 to 1.

The complete dataset consists of 2517 samples, distributed among three classes: 112 in the ageing class, 820 in the bearing class, and 1585 in the sliding class. The substantial imbalance in sample numbers across classes requires careful consideration during both training and testing phases. Additional information on how this imbalance is addressed will be outlined in the following section.

5. Methodology

5.1. Models

We employed three distinct NN architectures for our tasks, maintaining a consistent design for both anomaly classification and PI prediction. The models in consideration are fully-connected NNs featuring ReLU activation functions and comprised of two hidden layers. The initial set of models includes 32 hidden neurons in the first layer and 16 in the second layer. The second set has 64 in the first layer and 32 in the second layer, while the third set comprises 128 in the first layer and 64 in the second layer. It is important to highlight that the networks used for classification and those used for PI prediction differ in their output layers. The classification models have three outputs corresponding to the three classes of interest, while the PI prediction models possess a single output representing the predicted PI. We designate the regression models as EA_[n_1-n_2], where n_1 signifies the number of hidden neurons in the first layer, and n_2 denotes the same for the second layer. Similarly, the classification models follow a parallel naming convention and are denoted as EAC_[n_1-n_2].

To train the models for both regression and classification tasks, we followed a similar methodology. In both instances, we utilised pyNeVer[22], a comprehensive tool for managing, training, and verifying NNs. pyNeVer operates on the PyTorch[54] backend, offering users an intuitive custom training loop. Specifically, for both tasks, we employed the Adam optimizer with a learning rate set at 0.001 for 50 epochs. During training, we allocated 20% of the dataset for testing and 30% for validation. Additionally, a batch size of 16 was used for training, and 8 for validation. In the regression task, we adopted the standard mean square error (MSE) as both the training loss and the evaluation metric. For the classification task, the selection of the loss function involved more experimentation, as detailed in the previous section, owing to the significant class imbalance. To address this, we chose the weighted version of the Cross Entropy loss function, with weights inversely proportional to the number of samples in each class within the training set. We also implemented a procedure to ensure an equal representation of samples from each class in both the training and test sets. To evaluate the algorithm's performance on the test set, we computed accuracy for predicting each class individually and for all classes combined.

5.2. Properties

All our models underwent evaluation for similar local robustness properties. Specifically, for the classification models, we assessed their ability to withstand adversarial examples. Adversarial examples are inputs that have been subtly perturbed to mislead a machine learning model into making incorrect predictions. In our evaluation, this property was formally expressed as:

$$\forall x.\forall y.(|x - \hat{x}|_\infty \leq \varepsilon \wedge y = \nu(x)) \Rightarrow argmax(y) = \hat{y} \tag{1}$$

In simpler terms, this means that when a slight perturbation within the ε Chebyshev norm is applied to a given input sample $\hat{x}$, the NN must still predict the same class as it would for the original, unperturbed input.

For the regression models, we examined a related property:

$$\forall x.\forall y.(|x - \hat{x}|_\infty \leq \varepsilon \wedge y = \nu(x)) \Rightarrow |y - \hat{y}|_\infty < \delta \tag{2}$$

In this case, when the input is perturbed within the ε Chebyshev norm, the corresponding output must remain within a specified range bounded by the constant δ.

These formulations, however, are challenging to verify due to the universal quantification involved. Consequently, we reformulated the safety properties in their negated versions:

$$\exists x.\exists y.(|x - \hat{x}|_\infty \leq \varepsilon \wedge y = \nu(x)) \Rightarrow argmax(y) \neq \hat{y} \tag{3}$$

and

$$\exists x.\exists y.(|x - \hat{x}|_\infty \leq \varepsilon \wedge y = \nu(x)) \Rightarrow |y - \hat{y}|_\infty > \delta \tag{4}$$

If a verification tool can confirm that the negated versions of these properties do not hold, it certifies that the network under scrutiny adheres to the safety property.

6. Experimental Results

The experiments were carried out on a MacBook Air laptop, equipped with 24 GB of RAM and an Apple M2 CPU. The operating system utilised was macOS Sonoma 14.1.1, MPS was employed for training the NNs. For verifying the properties of interest, we utilised the pyNeVer verification tool, employing the over-approximate algorithm. The code necessary to reproduce our experiments is available at [55].

All NNs assessed in our experiments demonstrated adequate levels of precision for their respective tasks. In Table 1, we present the Mean Squared Error (MSE) calculated on the test set for the regression

Table 1

Summary of our experimental evaluation. **Model ID** corresponds to the identifier assigned to specific NNs. **Test Loss** indicates the Mean Squared Error (MSE) obtained by each model on the test set. **Epsilon** and **Delta** denote the values for ε and δ, respectively, considered for the property of interest. Lastly, **Result** and **Time** signify the outcome of the verification query and the time required (in seconds) by pyNeVer to resolve the query.

Model ID	Test Loss	Epsilon	Delta	Result	Time
EA_[32-16]	0.012	0.001	0.002	TRUE	11.56
		0.01	0.02	TRUE	11.52
		0.1	0.2	TRUE	11.55
EA_[64-32]	0.015	0.001	0.002	TRUE	13.81
		0.01	0.02	TRUE	12.88
		0.1	0.2	TRUE	13.86
EA_[128-64]	0.007	0.001	0.002	TRUE	14.79
		0.01	0.02	TRUE	16.30
		0.1	0.2	TRUE	27.09

Table 2

Summary of our experimental evaluation. **Model ID**, **Epsilon**, **Result**, and **Time** are the same of Table 1. **Accuracy** reports the percentage of samples classified correctly by each model on the test set. **SLD**, **BEA**, and **AGE** report, respectively, the percentage of samples belonging to the sliding, bearing, and ageing anomaly which are classified correctly by each model on the test set.

Model ID	Accuracy	SLD	BEA	AGE	Epsilon	Result	Time
EAC_[32-16]	92%	95%	95%	90%	0.001	FALSE	8.21
					0.01	FALSE	8.50
					0.1	TRUE	8.76
EAC_[64-32]	89%	86%	97%	85%	0.001	FALSE	10.01
					0.01	TRUE	10.57
					0.1	TRUE	13.08
EAC_[128-64]	97%	100%	100%	95%	0.001	FALSE	12.85
					0.01	FALSE	12.46
					0.1	TRUE	24.53

models under consideration. It is noteworthy that, even in the worst-case scenario, the MSE remains below 1.5×10^{-2}. Table 2 reports both the overall accuracy on the test set and the specific accuracy pertaining to each class. The relative accuracy signifies the proportion of correctly predicted samples belonging to a particular class among all the samples in that class. We introduced this additional performance metric to address the considerable class imbalance in the dataset. Interestingly, the complexity of the models does not necessarily correlate with their performance and precision. Even our smallest networks exhibited satisfactory performance for the given task. This emphasises the idea that, while larger NNs are gaining popularity, smaller models can still hold relevance in real-life industrial applications.

Regarding the formal verification of the models, Tables 1 and 2 illustrate that pyNeVer successfully verified the property of interest for all our models within a reasonable time frame. Notably, the size of the input space considered by the property appears to be as critical as the complexity of the models themselves in determining the verification time required by the tool. Additionally, the regression models seemed to be less robust to adversarial perturbations, as pyNeVer couldn't ensure the safety of any model. In other words, it consistently identified at least one data point that contradicted the safety property of interest. Conversely, the classification models exhibited greater resilience to adversarial perturbations, as pyNeVer was able to certify the safety of these models for various magnitudes of perturbations considered.

7. Conclusions and Future Work

In this work, we harnessed NN technology to tackle practical challenges in the elevator control domain, as part of the IMOCO4.E project. Our comprehensive approach covered both anomaly detection and magnitude prediction, employing fully-connected NNs for classification and regression tasks. Through experimentation, we showcased that even moderately-sized NN architectures delivered satisfactory precision and performance, emphasising the continued relevance of smaller models in practical industrial applications. Additionally, we explored the crucial aspect of local robustness, assessing the models' resistance to adversarial perturbations. Our findings, validated using the pyNeVer tool, underscored the tool's effectiveness in certifying safety properties for our models within reasonable time frames.

Our future endeavours will primarily concentrate on extending the experimental evaluation within the elevator control domain. We aim to enrich our dataset by introducing additional anomaly classes to the classification task, thereby augmenting the diversity and complexity of the data. This expansion will allow us to assess our models' performance across a broader spectrum of anomaly scenarios. To bolster the robustness of our regression models against adversarial perturbations, we plan to explore advanced techniques such as adversarial training, input preprocessing, and repair [56, 57, 58, 59, 60]. This initiative aims to further enhance the reliability and safety of our predictive maintenance systems for elevators in practical settings. By persistently refining our experiments and delving into these avenues, our goal is to offer more comprehensive solutions and insights for leveraging trustworthy NNs in elevator control and predictive maintenance.

Acknowledgments

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] D. Aineto, R. De Benedictis, M. Maratea, M. Mittelmann, G. Monaco, E. Scala, L. Serafini, I. Serina, F. Spegni, E. Tosello, A. Umbrico, M. Vallati (Eds.), Proceedings of the International Workshop on Artificial Intelligence for Climate Change, the Italian workshop on Planning and Scheduling, the RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and the Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (AI4CC-IPS-RCRA-SPIRIT 2024), co-located with 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[2] D. Guidotti, R. Masiero, L. Pandolfo, L. Pulina, Vector reconstruction error for anomaly detection: Preliminary results in the IMOCO4.E project, in: 28th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2023, Sinaia, Romania, September 12-15, 2023, IEEE, 2023, pp. 1–4. doi:10.1109/ETFA54631.2023.10275396.

[3] D. Guidotti, L. Pandolfo, L. Pulina, Detection of component degradation: A study on autoencoder-based approaches, in: 19th IEEE International Conference on e-Science, e-Science 2023, Limas-

sol, Cyprus, October 9-13, 2023, IEEE, 2023, pp. 1–2. doi:10.1109/E-SCIENCE58273.2023.10254890.

[4] C. Ferrari, M. N. Müller, N. Jovanovic, M. T. Vechev, Complete verification via multi-neuron relaxation guided branch-and-bound, in: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022, OpenReview.net, 2022.

[5] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, J. Z. Kolter, Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, Curran Associates, Inc., 2021, pp. 29909–29921.

[6] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljic, D. L. Dill, M. J. Kochenderfer, C. W. Barrett, The marabou framework for verification and analysis of deep neural networks, in: Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I, volume 11561 of *Lecture Notes in Computer Science*, Springer, Cham., 2019, pp. 443–452.

[7] S. Bak, H. Tran, K. Hobbs, T. T. Johnson, Improved geometric path enumeration for verifying relu neural networks, in: Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part I, volume 12224 of *Lecture Notes in Computer Science*, Springer, Cham., 2020, pp. 66–96.

[8] P. Kouvaros, T. Kyono, F. Leofante, A. Lomuscio, D. D. Margineantu, D. Osipychev, Y. Zheng, Formal analysis of neural network-based systems in the aircraft domain, in: Formal Methods - 24th International Symposium, FM 2021, Virtual Event, November 20-26, 2021, Proceedings, volume 13047 of *Lecture Notes in Computer Science*, Springer, Cham., 2021, pp. 730–740.

[9] G. Singh, T. Gehr, M. Püschel, M. T. Vechev, An abstract domain for certifying neural networks, Proc. ACM Program. Lang. 3 (2019) 41:1–41:30.

[10] H. Tran, X. Yang, D. M. Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, T. T. Johnson, NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems, in: Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part I, volume 12224 of *Lecture Notes in Computer Science*, Springer, Cham., 2020, pp. 3–17.

[11] P. Henriksen, A. R. Lomuscio, Efficient neural network verification via adaptive refinement and adversarial search, in: ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2513–2520.

[12] P. Henriksen, A. Lomuscio, DEEPSPLIT: an efficient splitting method for neural network verification via indirect effect analysis, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 2549–2555.

[13] D. Guidotti, L. Pandolfo, L. Pulina, Leveraging satisfiability modulo theory solvers for verification of neural networks in predictive maintenance applications, Inf. 14 (2023) 397. doi:10.3390/INFO14070397.

[14] S. Demarchi, D. Guidotti, A. Pitto, A. Tacchella, Formal verification of neural networks: A case study about adaptive cruise control, in: Proceedings of the 36th ECMS International Conference on Modelling and Simulation, ECMS 2022, Ålesund, Norway, May 30 - June 3, 2022, European Council for Modeling and Simulation, 2022, pp. 310–316. doi:10.7148/2022-0310.

[15] D. Guidotti, Enhancing neural networks through formal verification, in: Discussion and Doctoral Consortium papers of AI*IA 2019 - 18th International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, volume 2495 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 107–112.

[16] L. Pandolfo, L. Pulina, S. Vuotto, Smt-based consistency checking of configuration-based components specifications, IEEE Access 9 (2021) 83718–83726. URL: https://doi.org/10.1109/ACCESS.

2021.3085911. doi:10.1109/ACCESS.2021.3085911.

[17] R. Eramo, T. Fanni, D. Guidotti, L. Pandolfo, L. Pulina, K. Zedda, Verification of neural networks: Challenges and perspectives in the aidoart project (short paper), in: Proceedings of the 10th Italian workshop on Planning and Scheduling (IPS 2022), RCRA Incontri E Confronti (RiCeRcA 2022), and the workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (SPIRIT 2022) co-located with 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022), November 28 - December 2, 2022, University of Udine, Udine, Italy, volume 3345 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[18] D. Guidotti, Safety analysis of deep neural networks, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 4887–4888. doi:10.24963/IJCAI.2021/675.

[19] D. Guidotti, Verification and repair of neural networks, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 15714–15715. doi:10.1609/AAAI.V35I18.17854.

[20] D. Guidotti, F. Leofante, L. Pulina, A. Tacchella, Verification and repair of neural networks: A progress report on convolutional models, in: AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings, volume 11946 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 405–417. doi:10.1007/978-3-030-35166-3_29.

[21] D. Guidotti, F. Leofante, L. Pulina, A. Tacchella, Verification of neural networks: Enhancing scalability through pruning, in: ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2505–2512. doi:10.3233/FAIA200384.

[22] D. Guidotti, L. Pulina, A. Tacchella, pynever: A framework for learning and verification of neural networks, in: Automated Technology for Verification and Analysis - 19th International Symposium, ATVA 2021, Gold Coast, QLD, Australia, October 18-22, 2021, Proceedings, volume 12971 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 357–363. doi:10.1007/978-3-030-88885-5_23.

[23] S. Demarchi, D. Guidotti, Counter-example guided abstract refinement for verification of neural networks, in: Proceedings of the CPS Summer School PhD Workshop 2022 co-located with 4th Edition of the CPS Summer School (CPS 2022), Pula, Sardinia (Italy), September 19-23, 2022, volume 3252 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[24] D. Guidotti, Verification of neural networks for safety and security-critical domains, in: Proceedings of the 10th Italian workshop on Planning and Scheduling (IPS 2022), RCRA Incontri E Confronti (RiCeRcA 2022), and the workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (SPIRIT 2022) co-located with 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022), November 28 - December 2, 2022, University of Udine, Udine, Italy, volume 3345 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[25] D. Guidotti, L. Pandolfo, L. Pulina, Verifying neural networks with non-linear SMT solvers: a short status report, in: 35th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2023, Atlanta, GA, USA, November 6-8, 2023, IEEE, 2023, pp. 423–428. doi:10.1109/ICTAI59109.2023.00068.

[26] D. Guidotti, L. Pandolfo, L. Pulina, Verification of nns in the IMOCO4.E project: Preliminary results, in: 28th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2023, Sinaia, Romania, September 12-15, 2023, IEEE, 2023, pp. 1–4. doi:10.1109/ETFA54631.2023.10275345.

[27] D. Guidotti, L. Pandolfo, L. Pulina, Verifying neural networks with SMT: an experimental evaluation, in: 19th IEEE International Conference on e-Science, e-Science 2023, Limassol, Cyprus, October 9-13, 2023, IEEE, 2023, pp. 1–2. doi:10.1109/E-SCIENCE58273.2023.10254877.

[28] S. Demarchi, D. Guidotti, L. Pulina, A. Tacchella, Supporting standardization of neural networks verification with VNNLIB and coconet, in: Proceedings of the 6th Workshop on Formal Methods for ML-Enabled Autonomous Systems, FoMLAS@CAV 2023, Paris, France, July 17-18, 2023, volume 16 of *Kalpa Publications in Computing*, EasyChair, 2023, pp. 47–58. doi:10.29007/5PDH.

[29] D. Guidotti, L. Pandolfo, L. Pulina, Formal verification of neural networks: A "step zero" approach for vehicle detection, in: Advances and Trends in Artificial Intelligence. Theory and Applications - 37th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2024, Hradec Kralove, Czech Republic, July 10-12, 2024, Proceedings, volume 14748 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 297–309. doi:10.1007/978-981-97-4677-4_25.

[30] D. Guidotti, L. Pandolfo, L. Pulina, Verifying autoencoders for anomaly detection in predictive maintenance, in: Advances and Trends in Artificial Intelligence. Theory and Applications - 37th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2024, Hradec Kralove, Czech Republic, July 10-12, 2024, Proceedings, volume 14748 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 188–199. doi:10.1007/978-981-97-4677-4_16.

[31] D. Guidotti, F. Leofante, A. Tacchella, C. Castellini, Improving reliability of myocontrol using formal verification, IEEE Transactions on Neural Systems and Rehabilitation Engineering 27 (2019) 564–571. doi:10.1109/TNSRE.2019.2893152.

[32] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability, Comput. Sci. Rev. 37 (2020) 100270.

[33] F. Leofante, N. Narodytska, L. Pulina, A. Tacchella, Automated verification of neural networks: Advances, challenges and perspectives, CoRR abs/1805.09938 (2018).

[34] L. Pulina, A. Tacchella, An abstraction-refinement approach to verification of artificial neural networks, in: T. Touili, B. Cook, P. B. Jackson (Eds.), Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings, volume 6174 of *Lecture Notes in Computer Science*, Springer, Cham., 2010, pp. 243–257.

[35] R. Bunel, J. Lu, I. Turkaslan, P. H. S. Torr, P. Kohli, M. P. Kumar, Branch and bound for piecewise linear neural network verification, J. Mach. Learn. Res. 21 (2020) 42:1–42:39.

[36] S. Bak, nnenum: Verification of relu neural networks with optimized abstraction refinement, in: NASA Formal Methods - 13th International Symposium, NFM 2021, Virtual Event, May 24-28, 2021, Proceedings, volume 12673 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 19–36. doi:10.1007/978-3-030-76384-8_2.

[37] E. Wong, J. Z. Kolter, Provable defenses against adversarial examples via the convex outer adversarial polytope, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 5283–5292.

[38] E. Botoeva, P. Kouvaros, J. Kronqvist, A. Lomuscio, R. Misener, Efficient verification of relu-based neural networks via dependency analysis, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 3291–3299. doi:10.1609/AAAI.V34I04.5729.

[39] H. Zhang, T. Weng, P. Chen, C. Hsieh, L. Daniel, Efficient neural network robustness certification with general activation functions, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 4944–4953.

[40] T. Weng, H. Zhang, H. Chen, Z. Song, C. Hsieh, L. Daniel, D. S. Boning, I. S. Dhillon, Towards fast computation of certified robustness for relu networks, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 5273–5282.

[41] X. Huang, M. Kwiatkowska, S. Wang, M. Wu, Safety verification of deep neural networks, in: Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I, volume 10426 of *Lecture Notes in Computer Science*, Springer, Cham., 2017, pp. 3–29.

[42] M. Wu, M. Wicker, W. Ruan, X. Huang, M. Kwiatkowska, A game-based approximate verification of deep neural networks with provable guarantees, Theor. Comput. Sci. 807 (2020) 298–329.

[43] M. Wicker, X. Huang, M. Kwiatkowska, Feature-guided black-box safety testing of deep neural networks, in: Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part I, volume 10805 of *Lecture Notes in Computer Science*, Springer, Cham., 2018, pp. 408–426.

[44] W. Ruan, X. Huang, M. Kwiatkowska, Reachability analysis of deep neural networks with provable guarantees, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 2651–2659.

[45] W. Ruan, M. Wu, Y. Sun, X. Huang, D. Kroening, M. Kwiatkowska, Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 5944–5952.

[46] S. Kong, J. Ai, M. Lu, Y. Gong, GRAND: gan-based software runtime anomaly detection method using trace information, Neural Networks 169 (2024) 365–377. doi:10.1016/J.NEUNET.2023.10.036.

[47] L. Liu, L. Tian, Z. Kang, T. Wan, Spacecraft anomaly detection with attention temporal convolution networks, Neural Comput. Appl. 35 (2023) 9753–9761. doi:10.1007/S00521-023-08213-9.

[48] A. Minic, L. Jovanovic, N. Bacanin, C. Stoean, M. Zivkovic, P. Spalevic, A. Petrovic, M. Dobrojevic, R. Stoean, Applying recurrent neural networks for anomaly detection in electrocardiogram sensor data, Sensors 23 (2023) 9878. doi:10.3390/S23249878.

[49] C. Huang, J. Wen, Y. Xu, Q. Jiang, J. Yang, Y. Wang, D. Zhang, Self-supervised attentive generative adversarial networks for video anomaly detection, IEEE Trans. Neural Networks Learn. Syst. 34 (2023) 9389–9403. doi:10.1109/TNNLS.2022.3159538.

[50] X. Zhao, L. Zhang, Y. Cao, K. Jin, Y. Hou, Anomaly detection approach in industrial control systems based on measurement data, Inf. 13 (2022) 450. doi:10.3390/INFO13100450.

[51] D. Fährmann, N. Damer, F. Kirchbuchner, A. Kuijper, Lightweight long short-term memory variational auto-encoder for multivariate time series anomaly detection in industrial control systems, Sensors 22 (2022) 2886. doi:10.3390/S22082886.

[52] M. Cech, A. Beltman, K. Ozols, Digital twins and AI in smart motion control applications, in: 27th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2022, Stuttgart, Germany, September 6-9, 2022, IEEE, 2022, pp. 1–7.

[53] S. Mohamed, G. van der Veen, H. Kuppens, M. Vierimaa, T. Kanellos, H. Stoutjesdijk, R. Masiero, K. Määttä, J. W. van der Weit, G. Ribeiro, A. Bergmann, D. Colombo, J. Arenas, A. Keary, M. Goubej, B. Rouxel, P. Kilpeläinen, R. Kadikis, M. Armendia, P. Blaha, J. Stokkermans, M. Cech, A. Beltman, The IMOCO4.E reference framework for intelligent motion control systems, in: 28th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2023, Sinaia, Romania, September 12-15, 2023, IEEE, 2023, pp. 1–8. doi:10.1109/ETFA54631.2023.10275410.

[54] 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, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 8024–8035.

[55] D. Guidotti, RCRA-2024-IMOCO4E-UC1, https://github.com/darioguidotti/RCRA-2024-IMOCO4E-UC1, 2024. Accessed: 2024-09-06.

[56] P. Henriksen, F. Leofante, A. Lomuscio, Repairing misclassifications in neural networks using

limited data, in: SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, April 25 - 29, 2022, ACM, 2022, pp. 1031–1038.

[57] M. Sotoudeh, A. V. Thakur, Provable repair of deep neural networks, in: PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021, ACM, 2021, pp. 588–603.

[58] B. Goldberger, G. Katz, Y. Adi, J. Keshet, Minimal modifications of deep neural networks using verification, in: LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Alicante, Spain, May 22-27, 2020, volume 73 of *EPiC Series in Computing*, EasyChair, 2020, pp. 260–278.

[59] D. Guidotti, F. Leofante, C. Castellini, A. Tacchella, Repairing learned controllers with convex optimization: A case study, in: Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 16th International Conference, CPAIOR 2019, Thessaloniki, Greece, June 4-7, 2019, Proceedings, volume 11494 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 364–373. doi:10.1007/978-3-030-19212-9_24.

[60] D. Guidotti, F. Leofante, Repair of convolutional neural networks using convex optimization: Preliminary experiments, in: Proceedings of the Cyber-Physical Systems PhD Workshop 2019, an event held within the CPS Summer School "Designing Cyber-Physical Systems - From concepts to implementation", Alghero, Italy, September 23, 2019, volume 2457 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 18–28. URL: https://ceur-ws.org/Vol-2457/3.pdf.