Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P9555-cd

Graph Neural Networks for Anomaly Detection in Wind Turbines

Luca Pinciroli

Energy Department, Politecnico di Milano, Milan, Italy.. E-mail: luca.pinciroli@polimi.it

Piero Baraldi

Energy Department, Politecnico di Milano, Milan, Italy.. E-mail: piero.baraldi@polimi.it

Enrico Zio

MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France.
Energy Department, Politecnico di Milano, Milan, Italy.. E-mail: enrico.zio@polimi.it

The development of methods for anomaly detection in renewable energy systems is challenged by the complex spatio-temporal correlations among the measured signals, the variability of the operational and environmental conditions, and the presence of control systems that may hide anomalies in the signals patterns. To address these challenges, in this work we use Graph Neural Networks in identifying variations in the relationships among signals. Specifically, we propose a method which employs graph attention networks to dynamically consider interdependencies among signals, and gated recurrent units to capture temporal dependencies and system dynamics. The proposed method is compared to other state-of-the-art methods on a synthetic case study based on simulated wind turbine data. The obtained results show that the proposed method outperforms the comparison methods by more than 10% in terms of accuracy and reduces the detection delay by more than 50%.

Keywords: Renewable energy systems, Wind turbine, Anomaly detection, Graph neural networks.

1. Introduction

In the last decades, Renewable Energy Systems (RESs) have attracted increasing attention due to their capability to address the challenges related to environmental impact of energy production and reduce carbon emissions. By 2030, RESs are expected to account for nearly 50% of the global electricity mix (International Energy Agency (IEA) 2023). In this context, wind power, which in 2023 accounted for an installed capacity around 1TW (Global Wind Energy Council (GWEC) 2024), should be tripled by 2030 to effectively pursue the collective goal of the Paris Agreement (International Energy Agency (IEA). 2024). However, Wind Turbines (WTs) are subject to degradation and failure of their components, which requires that anomalies are promptly detected and maintenance interventions are timely planned (Panza, Pota, and Esposito 2023).

Deep Neural Networks (DNNs) have shown superior performances compared to traditional shallow neural networks in applications to anomaly detection in WTs (Pang et al. 2021). In (Rajaoarisoa et al. 2024), Autoencoders (AEs) were used to detect abnormal conditions of Wind Turbines (WTs); in (Lin et al. 2024) and (Lee et al. 2024), Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks were applied to anomaly detection of WTs using vibration signals; in (C. Zhang and Yang 2023), Generative Adversarial Networks (GANs) were combined with LSTM networks to detect pitting corrosion in the generator of WTs; in (F. Zhang et al. 2023), a combination of CNNs and LSTM networks was employed to detect anomalies on the main bearings of a WT.

Challenges hindering the widespread deployment of anomaly detection in WTs are: *i*) the interconnections among subsystems and components, which create complex spatio-temporal correlations among the signals and can lead to cascading propagation of anomalies; *ii*) the variability of the operational and environmental conditions, which influence and modify the systems behavior and the correlation among signals; *iii*) the presence of control systems that may hide anomalies in the measured signals by actuating on control variables; i*v*) the scarcity of data collected during anomalous conditions, which limits the applicability of supervised learning methods.

This work proposes a method for anomaly detection that addresses these challenges by using Graph Neural Networks (GNNs). A GNN is a type of neural network, specifically designed to deal with graph-structured data (Wu et al. 2022). GNNs have already been adopted for anomaly detection in industrial systems (Li et al. 2022; Jin et al. 2023). In (Feng, Liu, and Jiang 2023), an anomaly detection method based on the fusion of physical and statistical features was proposed; in (Miele, Bonacina, and Corsini 2022), a Graph Convolutional Autoencoder for Multivariate Time series (MTGCAE) was employed to model the sensor network of a WT as a dynamic functional graph and uses it to detect anomalies; in (W. Zhang, Zhang, and Tsung 2022), a Graph Relational Learning Network (GReLeN) based on variational autoencoders was used to construct a probabilistic relation graph. This graph was, then, employed for anomaly detection in water treatment and water distribution systems; in (Deng and Hooi 2021), a Graph Deviation Network (GDN) was developed to construct a graph of relationships between sensors and, then, use it to detect deviations from learnt patterns. Despite their satisfactory performances in the considered case studies, these methods only partially address the challenges of the variability of operational and environmental conditions, and do not explicitly consider the influence of control systems on the system behavior.

In this work, we develop a GNN model based on Graph ATtention networks (GATs) (Kipf and Welling 2016) for dynamically adjusting interdependencies and Gated Recurrent Units (GRUs) for long-term temporal modeling. The combination of GATs and GRUs allows enhancing the robustness of the anomaly detection method to the variability of operational and environmental conditions, and to the masking of anomalies by the control system. The model is trained using only normal condition data with the objective of reconstructing the signal values expected in normal conditions. Then, in operation, when the residual, i.e., the difference between signal measurements and reconstructions, is larger than a predefined threshold, the occurrence of an anomaly is detected. The effectiveness of the developed method is verified by its application to data simulated using the WT model presented in (Odgaard, Stoustrup, and Kinnaert 2013). The case study considers anomalies caused by failures of sensors, mechanical and electrical components in different operational and environmental conditions.

The structure of the work is as follows: Section 2 introduces the problem statement and formulation. Section 3 discusses the proposed method, providing details on graph neural networks and the proposed graph generation method. Section 4 presents the case study and Section 5 is dedicated to the analysis of the obtained results. Section 6 highlights the benefits and discusses the limitations of the proposed method. Finally, Section 7 concludes the work.

2. Problem Statement and Formulation

We consider a WT with a network of installed sensors that measures S physical quantities at each time step $t \in [1, T]$. These data are represented by the vector $\boldsymbol{x}^t = [x_1^t, \dots, x_s^t, \dots, x_S^t]$, where x_s^t is the measurement of the s^{th} sensor at time *t*. The objective of the present work is to develop a method that detect anomalies considering the evolution of the signal measurements $\boldsymbol{x}^t$ within a sliding time window [*t*-*W*+1, *t*] of length W.

To achieve this, we apply a sliding window of length W with stride D to generate $N = \frac{T-W}{D} + 1$ samples, denoted as $\boldsymbol{X}_n = [\boldsymbol{x}^{(n-1)D}, \dots, \boldsymbol{x}^{(n-1)D+W}]$, where $\boldsymbol{X}_n \in \mathbb{R}^{S \times W}$ for all $n \in [1, N]$. Each sample $\boldsymbol{X}_n$ is, then, used to construct a static graph $G_n = (V, E_n)$, where V and E_n represent the sets of vertices and edges, respectively.

Anomaly detection is performed by developing a signal reconstruction model, $f_\theta(\boldsymbol{X}_n, G_n)$, which reconstructs the signal values, $\boldsymbol{X}_n^\theta = f_\theta(\boldsymbol{X}_n, G_n)$, expected in normal conditions. If an anomaly occurs, the reconstruction error, $\epsilon_n = \boldsymbol{X}_n - \boldsymbol{X}_n^\theta$, $\epsilon_n \in \mathbb{R}^{S \times W}$, is expected to exceed a predefined threshold, K, for at least one of the S signals in at least one time step of the time window.

3. Proposed Method

Figure 1 shows the main steps of the proposed method, which are:

i. Signal denoising and normalization: signal measurements are denoised by applying a low-pass filter to remove high-frequency oscillations and normalized in the scale [0,1] to facilitate the evaluation of the correlations among them.

ii. Windowing: a sliding window of length W and stride D is used to generate the samples $\boldsymbol{X}_n$ from the multi-dimensional timeseries (Section 2). This allows focusing on local dynamics, enhancing the ability of the model to learn both temporal and spatial patterns.

iii. Graph generation: a graph $G_n = (V, E_n)$ is constructed from each sample $\boldsymbol{X}_n$. The set of vertices, V, is constituted by the measurements of the S sensors and the set of edges, E_n, and is defined on the basis of the characteristics of the sensor network and the physical relationships among the measured signals (Section 3.2). Binary values are used to indicate the absence (0) or presence (1) of an edge between two nodes.

iv. Model training: the available data collected in normal conditions and the corresponding graphs are used to train the model to reconstruct the measured signals;

v. Anomaly detection: the reconstruction model is applied to the test data. Then, the residuals, i.e. the difference between the measured and the reconstructed signals, are computed. If the residual of one signal is larger than a predefined threshold, an anomaly is detected.

3.1. *Graph neural networks*

A GNN is a neural network archetype for dealing with graph-structured data (Scarselli et al. 2009). It allows extracting spatial features, which are typically overlooked by other architectures (Li et al. 2022). The core working principle of GNNs consists in learning node representations by iteratively aggregating the information from node neighbors and combining it with the node own features. This process is repeated across multiple layers to better capture relationships between nodes also on broad graph structures.

Different GNN architectures have been proposed in literature for different purposes. Graph Convolutional Networks (GCN) (Kipf and Welling 2017) were proposed to adapt the notation of convolution to graph-structured data, GraphSAGE (SAmple and aggreGatE) was introduced to deal with the challenges of large-scale graphs (Hamilton, Ying, and Leskovec 2017), Graph Isomorphism Network (GIN) (Xu et al. 2019) were designed to maximize the expressiveness of GNNs and to distinguish more effectively different graph structures (Xu et al. 2019), and Graph Attention Networks (GAT) were proposed to introduce the self-attention mechanism in GNNs (Veličković et al. 2017).

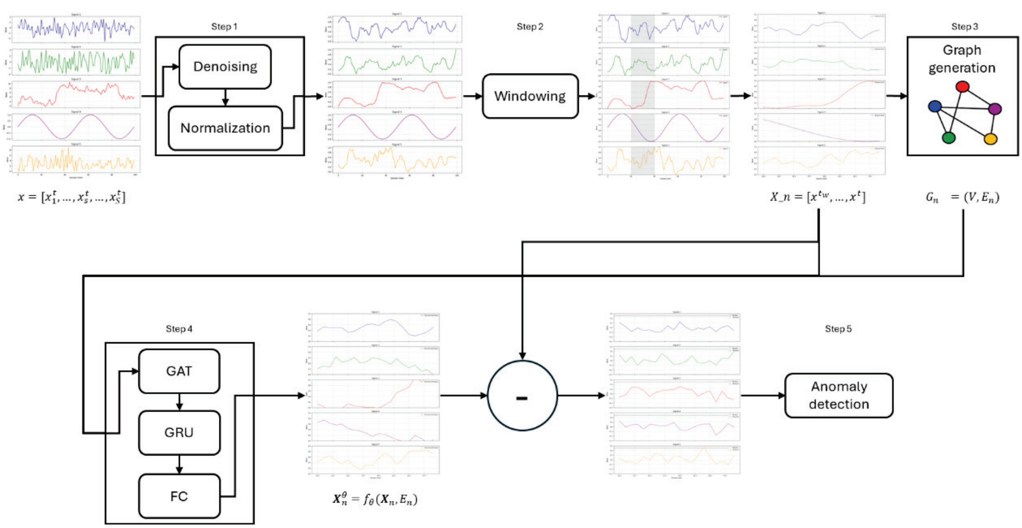

Figure 1. Overview of the proposed method.

In this work, we employ the GAT architecture since the self-attention mechanism enables dynamically learning attention coefficients, which automatically prioritize more relevant neighbors. This enhances the performance in scenarios where different neighbors contribute unequally to a node representation. More in details, the attention coefficient α_{ij}, representing the importance of the features of node j to node i, is computed as (Brody, Alon, and Yahav 2022):

$$\alpha_{ij} = \frac{\exp(\bar{a}^T LeakyReLU(\mathbf{\Theta}[\bar{h}_i || \bar{h}_j]))}{\sum_{k \in Z_i} \exp(\bar{a}^T LeakyReLU(\mathbf{\Theta}[\bar{h}_i || \bar{h}_k]))} \quad (1)$$

where $\bar{a}^T$ is a transposed learnable attention vector, $\mathbf{\Theta}$ is a learnable weight matrix, $\bar{h}_i || \bar{h}_j$ represents the concatenation of the features of node i and j, and Z_i is the neighborhood of node i. Additionally, attention heads can be used to enhance stability and expressiveness by computing the attention coefficients several times and aggregating information from multiple perspectives.

3.2. *Graph generation*

Different methods have been proposed to build graphs (Zhou et al. 2023). For example, K-Nearest Neighbors (KNN) and cosine similarity were used in (Li et al. 2022), and a domain-specific heuristic was explored in (Feng, Liu, and Jiang 2023).

In this work, we adopt a relationship-driven method in which edges are defined by a domain expert based on the analysis of the physical, functional and operational relationships among the signals. Specifically, relationships among signals are identified considering their measurement units, physical location, functional redundancy, causal relationships, dependence on operational and external conditions, and interactions. During graph generation, the adjacency matrix, A_n, representing the graph, $G_n = (V, E_n)$, is built. If a domain expert identifies a relationship between signals i and j, the element (*i,j*) of the adjacency matrix is set equal to 1, indicating that an edge connecting node i and node j exists; otherwise, the element (*i,j*) of the adjacency matrix is set equal to 0, indicating that the connecting edge does not exist. During training, the GAT layers compute the attention coefficients according to Eq.(1), determining the real importance of neighbor j for node i. Although the graph structure is static, i.e., the same structure is used for all samples, the importance of the different neighbors is computed dynamically by the GAT layers, allowing the model to dynamically adapt to signal variations.

4. Case Study

The proposed method is applied to a synthetic case study concerning anomaly detection in a three-bladed, pitch-controlled, variable-speed WT of nominal power $P_n = 4.8\,MW$. The dataset is generated using a modified version of the wind turbine model presented in (Odgaard, Stoustrup, and Kinnaert 2013), which allows simulating the measurements of $S = 15$ sensors when the WT is in different operational conditions. The list of the measured quantities is reported in Table 1 and the power curve of the considered WT is shown in Figure 2. The data are simulated in normal and anomalous conditions, with anomalies caused by $F = 10$ failure modes. The failure modes and the distributions of the parameters used for their simulation are reported in Table 2. In normal conditions, the damping factor, χ, the natural frequency, ω_n, and the drive train efficiency, η_{dt}, are set equal to $0.6\,\frac{Ns}{m}$, $11.11\,Hz$ and 0.97, respectively.

A total of $M = 200$ timeseries, each lasting $T = 4400s$ with a sampling frequency $f_s = 3\,Hz$, are simulated. This dataset comprises $M' = 100$ timeseries in normal conditions and $M'' = 100$ in anomalous conditions, with ten timeseries for each of the 10 failure modes. The duration of each failure is randomly sampled from a uniform distribution $U(134, 400)\,s$, and the failure starting time is randomly generated from a uniform distribution $U(0, 4000)\,s$.

The simulated signals are denoised by applying a low-pass filter with a cut-off frequency $f_{co} = 0.1\,Hz$. A sliding window of length $W = 134\,s$ and stride $D = 134\,s$ is applied to generate $N = 6600$ samples. Among these, $N' = 5919$ are from normal conditions and $N'' = 681$ are affected by one of the failure modes. The N' normal samples are separated among training validation and test sets. Specifically, $N_{tr} = 2640$ samples are used for training, $N_v = 660$ samples are used for validation and $N_{ts} = 2619$ samples are used for testing, whereas all the N'' anomalous samples are included in the test set. The adjacency matrix representing the graph defined by the domain expert is shown in Figure 3.

4.1. *Anomaly detection model training setup*

The proposed GNN-based model is composed of three GAT layers, three GRU layers and three Fully Connected (FC) layers. Each GAT layer is characterized by $h_{GAT} = 512$ hidden neurons and two attention heads. Each GRU layer includes $h_{GRU} = 1024$ hidden neurons, whereas the FC layers are characterized by $h_{FC} = 256$ hidden neurons. The ReLU (Rectified Linear Unit) is selected as activation function and dropout with probability $p_d = 0.1$ is used to prevent overfitting. The model is trained using the Adam optimizer with an initial learning rate of $\alpha = 0.001$ and a 10% decay every ten epochs. The batch is made of 32 samples and training is performed for a maximum of 1000 epochs with early stopping.

The experiments are repeated ten times considering different initializations and different subdivisions of the N' normal samples among training, validation and test sets. The model is implemented in python using PyTorch and PyTorch Geometric and the computation is performed on the CINECA's Leonardo supercomputer using a NVIDIA A100 SXM6 64GB GPU (Turisini, Cestari, and Amati 2024).

Table 1. List of signals and locations of the sensors.

Signal ID	Quantity	Location
1	Wind speed, v_w	Blades
2	Rotational speed, ω_{r_1}	Rotor
3	Rotational speed, ω_{r_2}	Rotor
4	Rotational speed, ω_{g_1}	Generator
5	Rotational speed, ω_{g_2}	Generator
6	Generator torque, τ_g	Generator
7	Produced power, P_g	Generator
8	Pitch position, $\beta_{1,1}$	Blade 1
9	Pitch position, $\beta_{1,2}$	Blade 1
10	Pitch position, $\beta_{2,1}$	Blade 2
11	Pitch position, $\beta_{2,2}$	Blade 2
12	Pitch position, $\beta_{3,1}$	Blade 3
13	Pitch position, $\beta_{3,2}$	Blade 3
14	Reference torque, τ_r	Generator
15	Reference pitch position, β_r	Blades

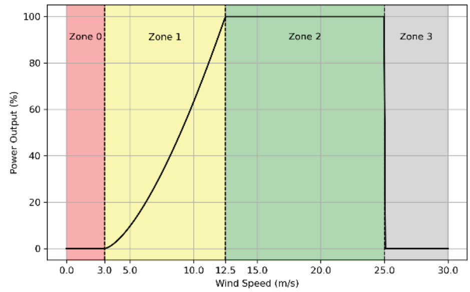

Figure 2. Reference power curve of the wind turbine.

Table 2. List of failure modes and description of their parameters.

Failure mode ID	Failure mode description	Parameter distribution
1	Signal 8 constant value, $\beta_{1,1} = k$	$k \sim N(5, 2)$
2	Signal 11 bias, $\beta_{2,2} = (1 + k)\beta_{2,2}$	$k \sim N(1.5, 2)$
3	Signal 12 constant value, $\beta_{3,1} = k$	$k \sim N(6, 2)$
4	Signal 2 constant value, $\omega_{r_1} = k$	$k \sim N(1.7, 2)$
5	Signal 3 and 4 gain factors increment, $\omega_{r_2} = k_1 \omega_{r_2}$, $\omega_{g_2} = k_2 \omega_{g_2}$	$k_1 \sim N(0.9, 1)$ $k_2 \sim N(0.5, 1)$
6	Pressure drop in pitch actuator 2, damping factor, $\chi = k_1$ and natural frequency, $\omega_n = k_2$	$k_1 \sim U(0.45, 0.75)$ $k_2 \sim U(5.73, 11.5)$
7	Air content in oil of pitch actuator 3, damping factor, $\chi = k_1$ and natural frequency, $\omega_n = k_2$	$k_1 \sim U(0.9, 1)$ $k_2 \sim U(3.42, 7.5)$
8	Sensor 6 offset, $\tau_g = k$	$k \sim N(-1000, 100)$
9	Increased friction of drive train, drive train efficiency reduction, $\eta_{dt} = \eta_{dt} - k$	$k \sim U(0.16, 0.42)$
10	Signal 4 constant value, $\omega_{g_1} = k$	$k \sim N(10, 3)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
7	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
8	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
9	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
10	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
11	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
12	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
13	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1
14	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1

Figure 3. Adjacency matrix.

5. Results

The performance of the proposed method is compared with that of the following state-of-the-art methods: AE, 1DCNN, LSTM networks and a combination of 1DCNN and LSTM networks. For all these methdos the hyperparameters have been optimized by trial-and-error on a validation set to minimize the reconstruction error. Two modified versions of the proposed method are also considered for comparison: *i*) the GAT layers are replaced by GCN layers to evaluate the importance of the attention mechanism and *ii*) the graph defined by the domain expert is replaced by a fully connected graph to evaluate the impact of experts' knowledge on the graph definition.

Figure 4 shows the Receiver Operating Characteristic (ROC) for the proposed method and the comparison methods. Notice that the proposed method achieves the most satisfactory performance for a large range of the threshold K. Table 3 reports the performance metrics obtained when the threshold for anomaly detection is 0.024, which is the minimum value for which less than 3% of the validation samples are misclassified as anomalies. The proposed method outperforms all comparison methods with respect to almost all metrics: it enhances the accuracy in the detection of anomalies (recall) by more than 10% with respect to the combination of 1DCNN and LSTM networks, and reduces the detection delay by 57% with respect to LSTM networks. The three GNN-based methods achieve the best results, highlighting that the introduction of the graph structure allows more effectively dealing with the interconnections among the components, the different operational conditions and the effect of control systems.

The use of GAT instead of GCN layers allows reducing both false and missed alarms, increasing both precision and recall. This result highlights the benefit of weighting neighbors, which allows prioritizing the most informative features. Also, the variability of the performance strongly increases when GCN are used instead of GAT. The use of a relationship-driven graph provided by the domain expert instead of a fully connected graph slightly impacts on accuracy, precision, recall and F1 score. This is because the GAT layers can autonomously reduce the weights of less important neighbors. However, notice that the detection delay is larger when using a fully connected graph structure, indicating that the attention mechanism is not able to completely remove non-relevant neighbors, which negatively impact on the overall performance.

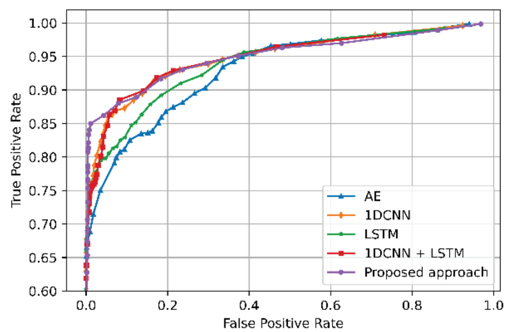

Figure 4. ROC curves of the proposed and comparison methods.

.

Table 3. Performance of the proposed and comparison methods. The most satisfactory performance is reported in bold.

Approach	Accuracy [%]	Precision [%]	Recall [%]	F1 score [%]	Detection delay [s]
AE	92,47 ± 0,16	**94,82 ± 0,79**	67,18 ± 0,30	78,64 ± 0,43	216,9 ± 1,4
1DCNN	90,75 ± 4,7	82,40 ± 13	74,93 ± 2,1	77,78 ± 6,6	149,7 ± 19,0
LSTM	92,80 ± 0,74	89,79 ± 3,9	73,67 ± 0,61	80,89 ± 1,4	149,6 ± 32,9
1DCNN+LSTM	93,18 ± 0,37	89,49 ± 0,71	75,11± 0,46	81,67 ± 0,45	157,1 ± 10,7
GCN+GRU+FC	<u>94,58 ± 0,92</u>	82,46 ± 14,36	82,37 ± 14,03	82,38 ± 14,09	<u>77,2 ± 16,9</u>
GAT+GRU+FC (fully connected graph)	94,31 ± 0,78	87,69 ± 3,54	<u>84,45 ± 1,02</u>	<u>85,99 ± 1,55</u>	193,5 ± 29,8
GAT+GRU+FC (proposed method)	**95,10 ± 0,41**	<u>90,92 ± 1,6</u>	**84,73 ± 0,72**	**87,71 ± 0,95**	**64,0 ± 8,9**

Also, the training time is longer when using a fully connected graph since more computations have to be performed.

6. Discussion

The obtained results show that incorporating domain expertise into the graph construction allows significantly reducing false and missed alarms. Also, the results suggest that GAT layers, are effective in capturing the system behavior, as they prioritize the most informative neighbors.

The main limitations the proposed method are: *i*) it relies on domain expertise to define the graph structure, which can become challenging when scaling to more complex systems, since omitting critical connections may negatively impact performance; *ii*) the computational cost of training GNN-based models, which is typically significantly larger than that of traditional methods and can hinder the application of the proposed method to more complex industrial systems.

7. Conclusions

A GNN-based method for anomaly detection in WTs has been proposed, aiming at improving detection performance of traditional AI-based methods in the common situations of interconnections among subsystems and components, variability of operational and environmental conditions and presence of control systems. The proposed method is based on a combination of GAT layers, used to dynamically consider interdependencies among signals, and GRUs, used to capture long-term temporal dependencies and variations in the system dynamics. The proposed method has been shown to outperform state-of-the-art methods in terms of accuracy, recall, F1 score and detection delay. Also, the comparison with two modified versions of the proposed method has highlighted the significant contribution to the overall anomaly detection results of using GAT and a relationship-driven graph provided by a domain expert.

Future work will focus on: *i*) enhancing the graph generation process to reduce reliance on expert knowledge, potentially through automated graph learning techniques; *ii*) validating the effectiveness and applicability of the proposed method considering a real-world case study.

Acknowledgement

The participation of Luca Pinciroli and Enrico Zio to this work is supported by the SAFEPOWER project under the HORIZON-CL5-2024-D3-01 call of the European Union, Grant Agreement 101172940. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.
The participation of Piero Baraldi to this work is supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence).

References

Brody, Shaked, Uri Alon, and Eran Yahav. 2022. "How Attentive Are Graph Attention Networks?" In *The Tenth International Conference on Learning Representations (ICLR 2022)*. http://arxiv.org/abs/2105.14491.

Deng, Ailin, and Bryan Hooi. 2021. "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series." In . http://arxiv.org/abs/2106.06947.

Feng, Chenlong, Chao Liu, and Dongxiang Jiang. 2023. "Unsupervised Anomaly Detection Using Graph Neural Networks Integrated with Physical-Statistical Feature Fusion and Local-Global Learning." *Renewable Energy* 206 (April):309–23. https://doi.org/10.1016/j.renene.2023.02.053.

Hamilton, William L., Rex Ying, and Jure Leskovec. 2017. "Inductive Representation Learning on Large Graphs." In *The Thirty-First Annual Conference on Neural Information Processing Systems (NIPS)*. http://arxiv.org/abs/1706.02216.

Jin, Ming, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. 2023. "A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection," July. http://arxiv.org/abs/2307.03759.

Kipf, Thomas N., and Max Welling. 2016. "Semi-Supervised Classification with Graph Convolutional Networks," September. http://arxiv.org/abs/1609.02907.

Kipf, Thomas N, and Max Welling. 2017. "Semi-Supervised Classification with Graph Neural Networks." In *Proceedings of 5th International Conference on Learning Representations, ICLR 2017*. Toulon.

Lee, Younjeong, Chanho Park, Namji Kim, Jisu Ahn, and Jongpil Jeong. 2024. "LSTM-Autoencoder Based Anomaly Detection Using Vibration Data of Wind Turbines." *Sensors* 24 (9). https://doi.org/10.3390/s24092833.

Li, Tianfu, Zheng Zhou, Sinan Li, Chuang Sun, Ruqiang Yan, and Xuefeng Chen. 2022. "The Emerging Graph Neural Networks for Intelligent Fault Diagnostics and Prognostics: A Guideline and a Benchmark Study." *Mechanical Systems and Signal Processing* 168 (April). https://doi.org/10.1016/j.ymssp.2021.108653.

Lin, Kuigeng, Jianing Pan, Yibo Xi, Zhenyu Wang, and Jianqun Jiang. 2024. "Vibration Anomaly Detection of Wind Turbine Based on Temporal Convolutional Network and Support Vector Data Description." *Engineering Structures* 306 (May). https://doi.org/10.1016/j.engstruct.2024.117848.

Miele, Eric Stefan, Fabrizio Bonacina, and Alessandro Corsini. 2022. "Deep Anomaly Detection in Horizontal Axis Wind Turbines Using Graph Convolutional Autoencoders for Multivariate Time Series." *Energy and AI* 8 (May). https://doi.org/10.1016/j.egyai.2022.100145.

Odgaard, Peter Fogh, Jakob Stoustrup, and Michel Kinnaert. 2013. "Fault-Tolerant Control of Wind Turbines: A Benchmark Model." *IEEE Transactions on Control Systems Technology* 21 (4): 1168–82. https://doi.org/10.1109/TCST.2013.2259235.

Panza, Maria Antonietta, Marco Pota, and Massimo Esposito. 2023. "Anomaly Detection Methods for Industrial Applications: A Comparative Study." *Electronics (Switzerland)* 12 (18). https://doi.org/10.3390/electronics12183971.

Rajaoarisoa, Lala, Michał Kuk, Szymon Bobek, and Moamar Sayed-Mouchaweh. 2024. "Hybrid and Co-Learning Approach for Anomalies Prediction and Explanation of Wind Turbine Systems." *Engineering Applications of Artificial Intelligence* 133 (July). https://doi.org/10.1016/j.engappai.2024.108046.

Scarselli, Franco, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. "The Graph Neural Network Model." *IEEE Transactions on Neural Networks* 20 (1): 61–80. https://doi.org/10.1109/TNN.2008.2005605.

Turisini, Matteo, Mirko Cestari, and Giorgio Amati. 2024. "LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI Applications." *Journal of Large-Scale Research Facilities JLSRF* 9 (1). https://doi.org/10.17815/jlsrf-8-186.

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. "Graph Attention Networks." In *The Sixth International Conference on Learning Representations (ICLR 2018)*. Vancouver. http://arxiv.org/abs/1710.10903.

Wu, Lingfei, Peng Cui, Jian Pei, and Liang Zhao. 2022. *Graph Neural Networks Foundations, Frontiers, and Applications*. https://doi.org/https://doi.org/10.1007/978-981-16-6054-2.

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. "How Powerful Are Graph Neural Networks?" In *The Seventh International Conference on Learning Representations (ICLR 2019)*.

Zhang, Chen, and Tao Yang. 2023. "Anomaly Detection for Wind Turbines Using Long Short-Term Memory-Based Variational Autoencoder Wasserstein Generation Adversarial Network under Semi-Supervised Training." *Energies* 16 (19). https://doi.org/10.3390/en16197008.

Zhang, Fanghong, Yuze Zhu, Chuanjiang Zhang, Peng Yu, and Qingan Li. 2023. "Abnormality Detection Method for Wind Turbine Bearings Based on CNN-LSTM." *Energies* 16 (7). https://doi.org/10.3390/en16073291.

Zhang, Weiqi, Chen Zhang, and Fugee Tsung. 2022. "GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning."

Zhou, Zhiyao, Sheng Zhou, Bochao Mao, Xuanyi Zhou, Jiawei Chen, Qiaoyu Tan, Daochen Zha, Yan Feng, Chun Chen, and Can Wang. 2023. "OpenGSL: A Comprehensive Benchmark for Graph Structure Learning." In *NeurIPS 2023, the Thirty-Seventh Annual Conference on Neural Information Processing Systems*. http://arxiv.org/abs/2306.10280.