Interpreting Outliers in Time Series Data through Decoding Autoencoder

Patrick Knab[1,2], Sascha Marton[1], Christian Bartelt[1] and Robert Fuder[2]

[1]*University of Mannheim, Germany*

[2]*Robert Bosch GmbH, Bühl, Germany*

Abstract

Outlier detection is a crucial analytical tool in various fields. In critical systems like manufacturing, malfunctioning outlier detection can be costly and safety-critical. Therefore, there is a significant need for explainable artificial intelligence (XAI) when deploying opaque models in such environments. This study focuses on manufacturing time series data from a German automotive supply industry. We utilize autoencoders to compress the entire time series and then apply anomaly detection techniques to its latent features. For outlier interpretation, we *i)* adopt widely used XAI techniques to the autoencoder's encoder. Additionally, *ii)* we propose **AEE**, **A**ggregated **E**xplanatory **E**nsemble, a novel approach that fuses explanations of multiple XAI techniques into a single, more expressive interpretation. For evaluation of explanations, *iii)* we propose a technique to measure the quality of encoder explanations quantitatively. Furthermore, we qualitatively assess the effectiveness of outlier explanations with domain expertise.

Keywords

Explainable Artificial Intelligence (XAI), Outlier Detection, Autoencoder

1. Introduction

Outliers represent exceptional instances that differ from a normal data distribution [1]. Artificial intelligence (AI) is pivotal in outlier (anomaly) detection applications, particularly in domains with high-dimensional data, such as time series. By analyzing patterns, trends, and dependencies, algorithms can effectively identify outliers and anomalous events in various domains, ranging from finance and healthcare to industrial processes [2, 1, 3]. In particular, manufacturing processes generate vast amounts of time series data, making timely and accurate outlier detection critical for maintaining operational efficiency and safety. However, opaque neural networks (NN) often lack the interpretability necessary for high-stakes environments [4, 3]. Consequently, explaining the model's decisions through explainable artificial intelligence (XAI) is essential to provide transparency and foster trust in automated decision-making [5, 6, 7].

This work utilizes convolutional autoencoders (CAE) to compress univariate time series data for anomaly detection in an automotive manufacturing plant. A complete time series is considered an outlier if the entire sequence deviates from the expected pattern. The purpose of utilizing CAE is to learn specific manufacturing process features and map a time series into a low-dimensional space at its bottleneck. An unsupervised anomaly detection algorithm then uses these latent features to identify outliers. Therefore, we are interested in explaining how the
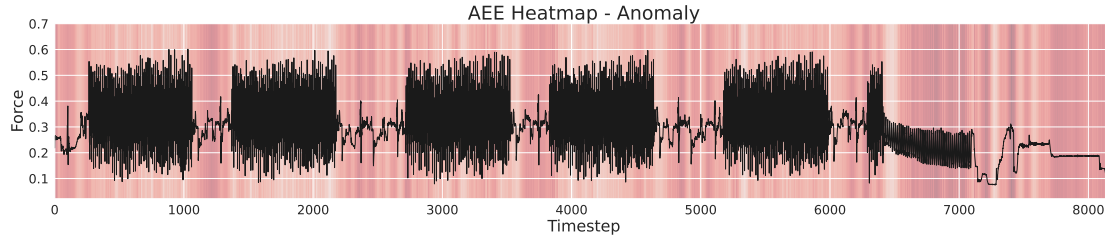

Figure 1: Aggregated Explanatory Ensemble - AEE. The aggregated explanation is represented by a heat map in the background, with deeper shades of red indicating areas of greater significance for its explanation in the time series. The black curve visualizes an anomalous time series, with the explanation highlighting a disruption in the pattern between the 6300 and 7200 marks in the time series.

encoder transformation contributes to outlier detection by employing established XAI methods (Section 2) like Grad-CAM [8], LIME [9], SHAP [10], and LRP [11, 12, 13], since we use the CAE's latent features for detecting outliers. The explanations of the XAI techniques mentioned above lead to fluctuations due to their unique features. This diversity motivates us to combine these explanations into a single, more comprehensive one: **AEE - Aggregated Explanatory Ensemble** (see Section 3.2), visualized in Figure 1 for an anomalous time series instance. Since ground-truth data for evaluating the produced explanations are often missing, counterfactuals are widely recognized as an effective quantitative method for XAI techniques [14, 15]. In our work, we implement a revised version of the quality measurement (QM) procedure, originally proposed in [16], as detailed in Section 3.3. We assess the effectiveness of the techniques both qualitatively and quantitatively (see Section 4) based on the underlying manufacturing process. Our primary focus is on discussing their implications for erroneous time series data to gain insights into the property of being an outlier as a complete time series.

2. Related XAI Approaches

The following section briefly mentions the XAI approaches used in this work. They were chosen for their well-known status and ability to cover explainability from different aspects, e.g., local vs. global and model-agnostic vs. model-specific explanations. These techniques share the goal of providing post hoc explanations, but each employs different approaches to achieve explainability.

CAM (Class Activation Mapping), proposed by Zhou et al. [17], is a local and model-specific technique for explaining convolutional neural networks (CNN). Selvaraju et al. [8] enhanced this approach with Grad-CAM, incorporating gradients into the explanation process. This improvement removes the requirement for a global average pooling layer, making the method applicable to a broader range of model architectures.

LIME (Local Interpretable Model-agnostic Explanations), proposed by Ribeiro et al. [9], is another well-known local but model-agnostic technique. It achieves interpretability by locally approximating the behavior of a complex neural network with an interpretable machine learning algorithm.

SHAP (SHapley Additive exPlanations) is a game theory-inspired model-agnostic technique

proposed by Lundberg et al. [10], which can provide local or global explanations. It computes Shapley values for the model's data inputs, providing a global explanation that considers multiple data instances simultaneously. Studies carried out in [4, 3] utilized SHAP to unravel the inner workings of a complete autoencoder.

LRP (Layer-wise Relevance Propagation) aims at comprehending a model's inner workings by retroactively propagating the internal values of its layers. Following this strategy, LRP seeks to assign relevance scores to different input features [11, 12, 13].

3. Application of XAI to Autoencoder

Notation. An univariate time series instance $\mathbf{t}$ is fed into the convolutional autoencoder via the function $\hat{\mathbf{t}} = D(E(\mathbf{t}))$, with encoder E, decoder D, and latent space L, where $L = E(\mathbf{t})$. The output $\hat{\mathbf{t}}$ is a reconstruction of $\mathbf{t}$. We denote explanation $\mathbf{E}$ as the output of an XAI technique for a time series $\mathbf{t}$.

3.1. Adapting XAI Techniques to Encoder

We employ a 1D convolutional autoencoder (1D CAE) to reduce feature dimensions and detect anomalies in time series data (see Section 4.1 for more details). We apply XAI techniques to the encoder since we use its output for anomaly detection. While the straightforward architecture facilitates the application of the XAI methods introduced in Section 2, their utilization, although widely applied in diverse machine learning scenarios, remains relatively limited within the realm of 1D CAE, mainly when applied to time series data. We adapt these methods to improve their capability to provide 1D explanations for 1D convolutional networks in the form of heatmaps. The application of the XAI methods above yields two distinct types of explanations:

- **Individual Feature Explanation:** For each latent feature, $l_i \in L$ (where i is the feature index), we generate a dedicated heatmap. This allows us to inspect how individual features in $\mathbf{t}$ contribute to the reconstruction process (see Appendix A).
- **Combined Feature Explanation:** In addition to the individual views, we also create a unified heatmap that integrates all latent features into a single representation (see Figure 2a). This combined view provides a holistic understanding of how the interplay between features in $\mathbf{t}$ influences the reconstruction process. All experiments and figures in this paper use combined feature explanations.

3.2. AEE - Aggregated Explanatory Ensemble.

With the application of the covered XAI approaches (see Section 2), we generate a set of diverse explanations. Each XAI technique provides distinct insights: Grad-CAM emphasizes spatial relevance, LIME offers local interpretability, SHAP delivers global explanations, and LRP traces relevance propagation (see Figure 2). By aggregating these methods, AEE leverages their strengths for a holistic understanding of anomalies. We restrict AEE to a time series $\mathbf{t}$ that stores the diverse explanations $\mathbf{E}_i$ in an array $\mathbf{E}_i^x$, where i indicates the index of $\mathbf{t}$ and x denotes the underlying XAI technique. To ensure equal consideration for each explanation, we individually

scale each element $\mathbf{E}_i^x$ based on its importance scores. Mathematically, the scaled explanation $\mathbf{SE}_i^x$ is given by:

$$\mathbf{SE}_i^x = \left(\frac{\mathbf{E}_i^x - \min(\mathbf{E}^x)}{\max(\mathbf{E}^x) - \min(\mathbf{E}^x)} \right) \times (a_{\max} - a_{\min}) + a_{\min}. \tag{1}$$

Here, $a_{\min}$ and $a_{\max}$ are the minimum and maximum values desired for scaling $\mathbf{SE}_i^x$. After scaling, we compute the mean value for each point i on the X axis. We denote the aggregated version as $\mathbf{A}_i$, where i represents the aggregated value for the ith point of the time series $\mathbf{t}$ on the X axis, mathematically:

$$\mathbf{A}_i = \frac{1}{|x|} \sum_{j=1}^{|x|} \mathbf{SE}_i^j. \tag{2}$$

Here, $|x|$ denotes the count of explanations stored in $\mathbf{SE}_i^x$. Alternatively, a weighting scheme can be employed instead of equal contribution to assign more relevance to specific explanations.

3.3. Quality Measurement of Encoder's Explanation

Given the interpretability constraints of the XAI results [5], we quantitatively analyze the explanations generated by each method using a modified version of the quality measurement function proposed by Schlegel et al. [16]. In this work, the XAI techniques focus on the encoder's explainability, resulting in a multi-regression task. Using the reconstruction error as a quality measurement would involve the decoder, misleading the measurement of the encoder's explanation. Instead, we aim to analyze the projections of the original time series $\mathbf{t}$, a randomly perturbed version $\mathbf{t_r^c}$, and a version perturbed based on explanation results $\mathbf{t^c}$ in the latent space. This approach operates independently of the decoder, focusing on explaining the techniques applied to the encoder. Adversarial perturbations [18], which manipulate predictions, suggest that the distance between $\mathbf{t_r^c}$ and $\mathbf{t}$ should be smaller than between $\mathbf{t}$ and $\mathbf{t^c}$. Thus, we define the quality measurement for the encoder as:

$$qm_e(\mathbf{t}, \mathbf{t}) \leq qm_e(\mathbf{t}, \mathbf{t_r^c}) \leq qm_e(\mathbf{t}, \mathbf{t^c}). \tag{3}$$

Here, qm_e measures the Euclidean distance between the original and perturbed time series in the latent space. The underlying theory is that perturbations based on explanation results have a more significant impact on the model's predictions than random noise [14, 15]. The approach applies to individual and combined feature explanations, revealing the importance of features for the outlierness property of the instance.

4. Experimentation

4.1. Experimental Setup

Dataset. As introduced in Section 1, our demonstration employs univariate time series data originating from a production plant. More specifically, it covers one process in a manufacturing line consisting of multiple processes. The dataset consists of 18,412 time series instances, each containing 8,192 data points. The test station (end of the line) automatically labels the data to

Table 1
Anomaly Detection Performance Measurements. The table contains the precision, recall, and F1-score performance metrics of the developed anomaly detection pipeline for the test set.

Class	Precision	Recall	F1-Score	Support
0 (OK)	1.00	1.00	1.00	5441
1 (NOK)	0.89	0.63	0.74	38

indicate normal operation (OK) or an error (not OK—NOK) during production; NOK accounts for 0.68% overall. This information can be used to assess whether the found outliers correspond to actual errors identified by the test station.

Anomaly Detection Pipeline. Our 1D CAE architecture comprises three convolutional layers with ReLU activation functions, followed by max-pooling layers and a bottleneck layer with a three-dimensional latent space (see Appendix B). We divide the data into three sets to train the AE: a training set for model training (0.66% NOK), a validation set (0.74% NOK), and a separate set for testing (0.70% NOK) the model's performance.

We intentionally include known anomalies in the training process, as instances with NOK labels may contain errors originating from other processes in the manufacturing line that the time series does not cover. In addition, the proportion of abnormal instances is low enough (less than 1%) that the autoencoder continues to learn to reconstruct the time series correctly without learning anomalies. The pipeline consists of an anomaly detection mechanism that utilizes the latent feature space as input (see Appendix C). Specifically, we employ the density-based spatial clustering of applications with noise (DBSCAN) algorithm [19]. Table 1 presents the performance metrics of the anomaly detection pipeline, categorized into NOK and OK classes. These results are based on the evaluation of the test dataset.

4.2. Qualitative Evaluation - Anomaly Interpretation

In the following, we discuss the utility of XAI techniques to interpret the encoder, focusing on understanding why specific instances lead to anomalies by leveraging domain-specific knowledge of the underlying manufacturing process. We examine an exemplary time series classified as NOK for all explanation techniques in Figure 2 and Figure 3. The illustrative case diverges notably in its final third segment, as the pattern is expected to exhibit distinct characteristics compared to the preceding two thirds of the time series (see Appendix D). We explicitly demonstrate the anomalies shown here using examples that are easy to visually understand as outsiders.

We initiate with Grad-CAM (Figure 2a), revealing a heatmap that distinctly accentuates positions later in **t**, precisely aligning with observable areas of technical failures in the manufacturing process. This targeted explanation effectively identifies the specific region preceding real-world anomalies. Subsequently, LIME (Figure 2b) highlights the same area as CAM, but its interpretation is more straightforward because of its apparent intensity. Moreover, it also subtly indicates regions in intermediate areas of the time series. SHAP (Figure 2c) pinpoints the same critical area of primary importance, consistent with the findings of the previous methods. Compared to the preceding, the final standalone method LRP (Figure 2d) diverges
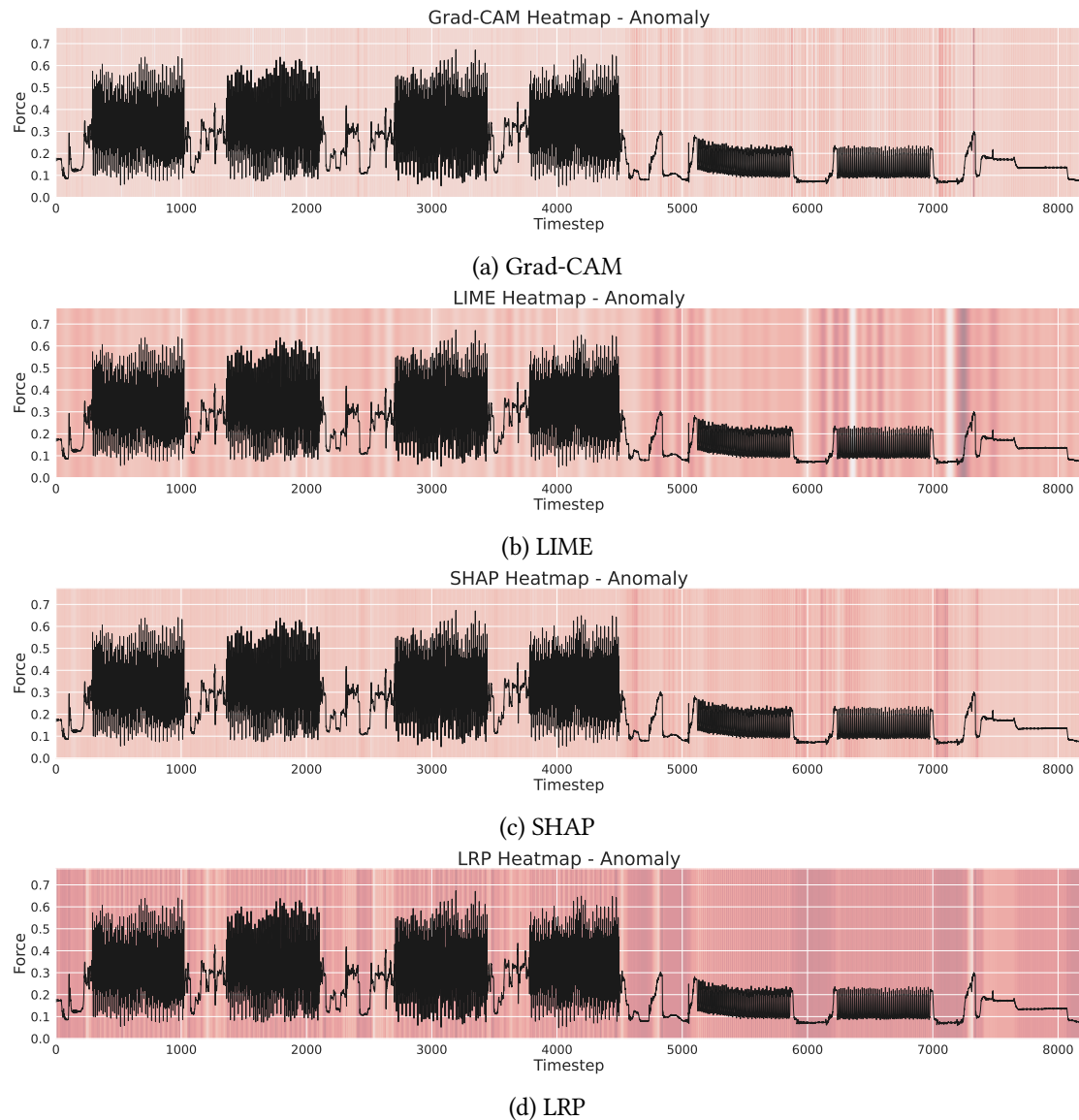
(a) Grad-CAM

(b) LIME

(c) SHAP

(d) LRP

Figure 2: Individual XAI Results. The XAI results are presented in the form of heatmaps. The black portions of the images denote the time series signal. These displayed instances were identified as abnormal by the AE's pipeline. The heatmap in the background indicates feature importance using varying intensities of red. We must evaluate color intensity individually as XAI techniques calculate feature importance differently.

in its explanation. Although it does not explicitly emphasize the most pronounced pattern, it assigns varying degrees of importance to different segments and provides valuable insights for manual analysis by a domain expert.

Figure 3 shows the aggregated explanation. Parallel to Grad-CAM and SHAP, the region signaling an abnormal pattern is precisely accentuated, and the aggregated version amplifies
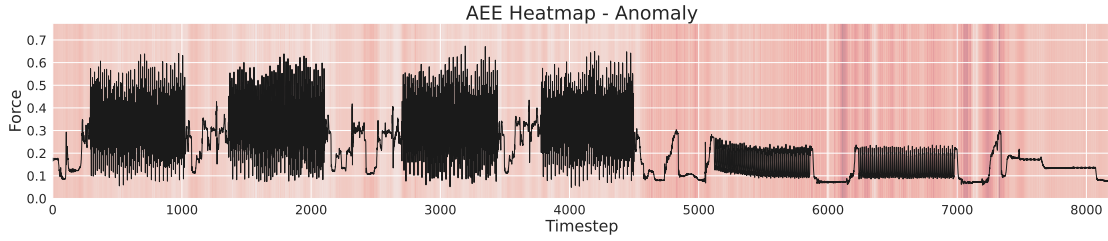
Figure 3: AEE XAI Results. This figure presents the results of the XAI analysis for the AEE approach. The format and layout of these explanations are consistent with those shown in Figure 2.

the color representation, enhancing interpretability. Besides confirming the importance of the known area, this approach offers additional insights into other parts of the time series, e.g., it prioritizes early regions that indicate possible technical abnormalities. Repeated experiments prove its explanations are more stable due to its aggregation property, mitigating the negative implications of instability [20].

4.3. Quantitative Evaluation - XAI Quality Measurement.

Figure 4 depicts the QM (normalized Euclidean distances) distributions, where boxes represent the interquartile range (IQR) from Q1 to Q3, with a median line (Q2). The fences extend ±1.5 times the IQR. The OK category includes 100 randomly selected instances, and the NOK category comprises 38. The noise/shuffle box (green) represents QM values $\mathbf{t_r^c}$, and the XAI box (red) represents $\mathbf{t^c}$.

The visualization indicates that each QM XAI score consistently outperforms its QM noise counterpart. The scores for the NOK cluster are significantly higher, demonstrating the effectiveness of using explanations for outlier interpretation. LRP and LIME overlap between Noise and XAI, while Grad-CAM and SHAP display a clearer separation in their explanations. The AEE produces a significant result, indicating that aggregating multiple explanations sharpens the distinction between relevant and irrelevant features within a time series, improving explanation quality.
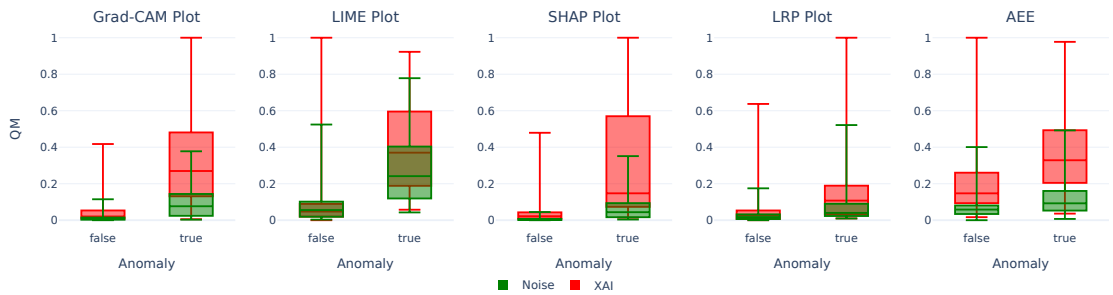

Figure 4: Interquartile Range Quality Measurements. The visualization depicts quality measurement scores for each XAI technique, categorized into true anomalies (NOK) and false anomalies (OK). The measurements are further stratified into noise ($\mathbf{t_r^c}$ - XAI shuffled), denoted by green, and XAI ($\mathbf{t^c}$ - XAI perturbated), represented by red.

4.4. Limitations and Future Work

Our study applied XAI techniques to CAE, leaving the potential for other architectures such as variational autoencoders (VAE) [21] and recurrent neural networks (RNN) [22, 23, 7] unexplored. Additionally, the evaluation of these techniques was primarily based on qualitative assessments, as anomalies required examination by domain experts. Future research on datasets not requiring expert knowledge should consider integrating additional quantitative methods to complement qualitative insights [24]. In addition, a clear distinction between explanation and interpretation should be established [25], recognizing that not all explanations are inherently human-interpretable [26], as it was sometimes the case in this scenario.

Furthermore, exploring different weighting schemes for AEE could enhance the interpretation and accuracy of feature importance calculations in various scenarios. We outlined the experimentation on the time series manufacturing use case. Future research could involve testing the AEE approach across various data types, such as images or text. Regarding XAI approaches, future work could focus on improving time series segmentation using foundation models [27], particularly beneficial for LIME. Another promising direction is to direct the explanations not toward the latent features themselves but the classes in the latent space that signify the presence or absence of anomalies. Lastly, extending this methodology to multivariate time series [28, 23, 29, 30] or even multimodal data [21] presents another intriguing avenue for future exploration.

5. Conclusion

This paper contributes to the application of XAI techniques to CAEs for analyzing outlier properties within the latent space of time series data in the operational context of a manufacturing plant. We employed well-established XAI methods to demonstrate the practicality and effectiveness of these techniques in interpreting outliers. In addition, we introduced AEE, an ensemble of multiple XAI techniques. We quantitatively evaluated the different explanations using a QM approach specifically modified to fit the encoder of an AE. Moreover, the application of XAI techniques provided explanations for these outliers, accurately highlighting the abnormal segments within the time series. This alignment confirms the utility of XAI in providing meaningful insights into anomalies and building confidence in the system through the interpretation of XAI results.

Acknowledgments

References

[1] A. Boukerche, L. Zheng, O. Alfandi, Outlier detection: Methods, models, and classification, ACM Comput. Surv. 53 (2020). URL: https://doi.org/10.1145/3381028. doi:10.1145/

`3381028`.

[2] T. Kieu, B. Yang, C. Guo, C. S. Jensen, Y. Zhao, F. Huang, K. Zheng, Robust and explainable autoencoders for unsupervised time series outlier detection—extended version, 2022. URL: https://arxiv.org/abs/2204.03341. doi:`10.48550/ARXIV.2204.03341`.

[3] L. Antwarg, R. M. Miller, B. Shapira, L. Rokach, Explaining anomalies detected by autoencoders using shapley additive explanations, Expert Systems with Applications 186 (2021) 115736. URL: https://www.sciencedirect.com/science/article/pii/S0957417421011155. doi:`https://doi.org/10.1016/j.eswa.2021.115736`.

[4] K. Roshan, A. Zafar, Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation(shap), CoRR abs/2112.08442 (2021). URL: https://arxiv.org/abs/2112.08442. `arXiv:2112.08442`.

[5] C. Molnar, Interpretable Machine Learning, 2019.

[6] T. K. K. Ho, N. Armanfard, Multivariate time-series anomaly detection with contaminated data, 2024. URL: https://arxiv.org/abs/2308.12563. `arXiv:2308.12563`.

[7] M. A. Belay, S. S. Blakseth, A. Rasheed, P. Salvo Rossi, Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions, Sensors 23 (2023). URL: https://www.mdpi.com/1424-8220/23/5/2844. doi:`10.3390/s23052844`.

[8] R. R. 2Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[9] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 1135–1144. URL: https://doi.org/10.1145/2939672.2939778. doi:`10.1145/2939672.2939778`.

[10] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774. URL: https://arxiv.org/abs/1705.07874. doi:`10.48550/ARXIV.1705.07874`.

[11] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proceedings of the IEEE 109 (2021) 247–278. doi:`10.1109/JPROC.2021.3060483`.

[12] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE 10 (2015) 1–46. URL: https://doi.org/10.1371/journal.pone.0130140. doi:`10.1371/journal.pone.0130140`.

[13] S. A. Siddiqui, D. Mercier, M. Munir, A. Dengel, S. Ahmed, TSViz: Demystification of deep learning models for time-series analysis, IEEE Access 7 (2019) 67027–67040. URL: https://doi.org/10.1109%2Faccess.2019.2912823. doi:`10.1109/access.2019.2912823`.

[14] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, S. Lee, Counterfactual visual explanations, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 2376–2384. URL: https://proceedings.mlr.press/v97/goyal19a.html.

[15] F. Bodria, R. Guidotti, F. Giannotti, D. Pedreschi, Interpretable latent space to enable counterfactual explanations, in: P. Pascal, D. Ienco (Eds.), Discovery Science, Springer Nature Switzerland, Cham, 2022, pp. 525–540.

[16] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, D. A. Keim, Towards a rigorous evaluation of XAI methods on time series, CoRR abs/1909.07082 (2019). URL: http://arxiv.org/abs/1909.07082. arXiv:1909.07082.

[17] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) abs/1512.04150 (2015). URL: http://arxiv.org/abs/1512.04150. arXiv:1512.04150.

[18] G. Fidel, R. Bitton, A. Shabtai, When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures, CoRR abs/1909.03418 (2019). URL: http://arxiv.org/abs/1909.03418. arXiv:1909.03418.

[19] D. Reynolds, Gaussian Mixture Models, Springer US, Boston, MA, 2009, pp. 659–663. URL: https://doi.org/10.1007/978-0-387-73003-5_196. doi:10.1007/978-0-387-73003-5_196.

[20] G. Visani, E. Bagli, F. Chesani, Optilime: Optimized LIME explanations for diagnostic computer algorithms, CoRR abs/2006.05714 (2020). URL: https://arxiv.org/abs/2006.05714. arXiv:2006.05714.

[21] D. Park, Y. Hoshi, C. C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics and Automation Letters 3 (2018) 1544–1551.

[22] O. I. Provotar, Y. M. Linder, M. M. Veres, Unsupervised anomaly detection in time series using lstm-based autoencoders, in: 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), 2019, pp. 513–517. doi:10.1109/ATIT49449.2019.9030505.

[23] H. Homayouni, S. Ghosh, I. Ray, S. Gondalia, J. Duggan, M. G. Kahn, An autocorrelation-based lstm-autoencoder for anomaly detection on time-series data, in: 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 5068–5077. doi:10.1109/BigData50022.2020.9378192.

[24] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, C. Seifert, From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3583558. doi:10.1145/3583558.

[25] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103. doi:https://doi.org/10.1016/j.inffus.2019.12.012.

[26] S. Mohseni, N. Zarei, E. D. Ragan, A multidisciplinary survey and framework for design and evaluation of explainable ai systems, ACM Trans. Interact. Intell. Syst. 11 (2021). URL: https://doi.org/10.1145/3387166. doi:10.1145/3387166.

[27] P. Knab, S. Marton, C. Bartelt, Dseg-lime: Improving image explanation by hierarchical data-driven segmentation, 2024. arXiv:2403.07733.

[28] T. Kieu, B. Yang, C. S. Jensen, Outlier detection for multidimensional time series using deep neural networks, in: 2018 19th IEEE International Conference on Mobile Data Management

(MDM), 2018, pp. 125–134. doi:`10.1109/MDM.2018.00029`.

[29] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 3395–3404. URL: https://doi.org/10.1145/3394486.3403392. doi:`10.1145/3394486.3403392`.

[30] H. Meng, C. Wagner, I. Triguero, Segal time series classification — stable explanations using a generative model and an adaptive weighting method for lime, Neural Networks 176 (2024) 106345. URL: https://www.sciencedirect.com/science/article/pii/S0893608024002697. doi:`https://doi.org/10.1016/j.neunet.2024.106345`.

A. Individual Feature Explanation

Figure 5 shows an instance that the pipeline classified as NOK, featuring the reconstructed time series in red and the original time series in black. The underlying explanation is provided through individual feature explanations, where a distinct heatmap visually explains each latent feature.
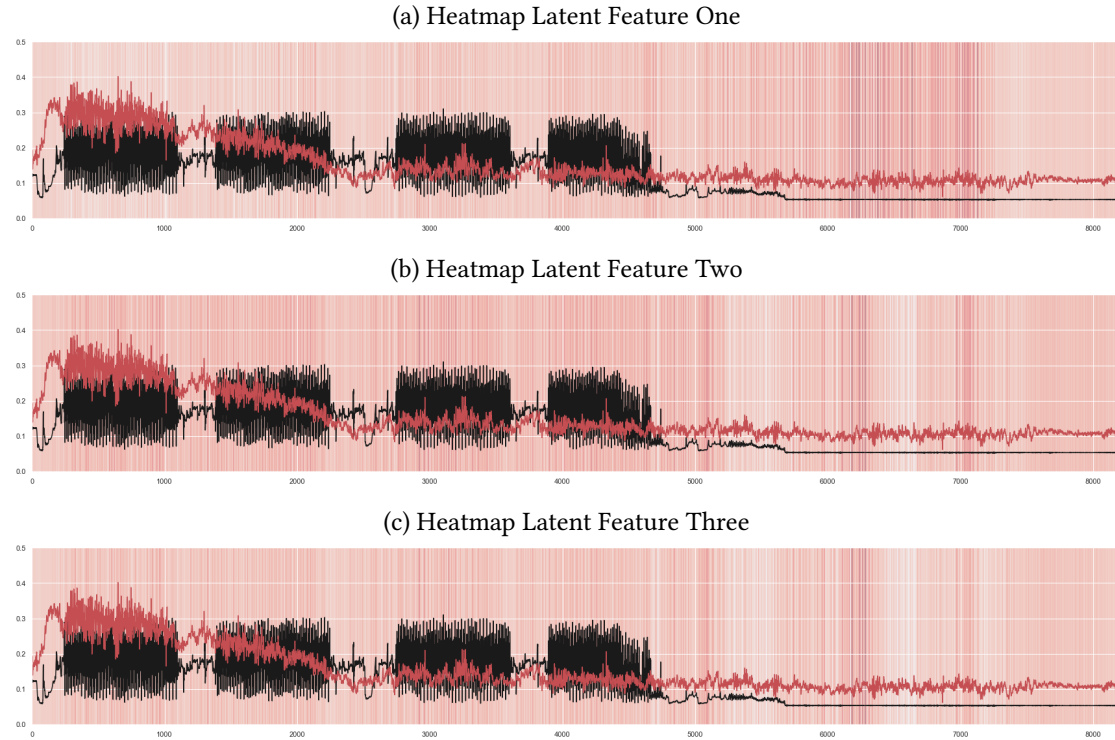
(a) Heatmap Latent Feature One

(b) Heatmap Latent Feature Two

(c) Heatmap Latent Feature Three

Figure 5: Grad-CAM - **Individual Feature Heatmaps.** The images illustrate individual latent feature explanations in the form of a heatmap generated by Grad-CAM. The black curve illustrates the original time series, while the red curve represents its reconstruction.

B. Autoencoder Architecture

In the following, we present the defined search space of hyperparameters for tuning an AE in this work. The search space has been explored by 100 runs and 500 epochs each. Figure 6 represents the building blocks we tuned during this process.

- The amount of **CNN blocks** consists of an optional dropout and max pooling layer. We restrict this size to at least one and a maximum of three blocks. This number applies to both the encoder and the decoder.
- In contrast to the number of CNN blocks, this number varies for **DNN blocks** between the encoder and decoder parts. Both can have up to two DNN blocks.
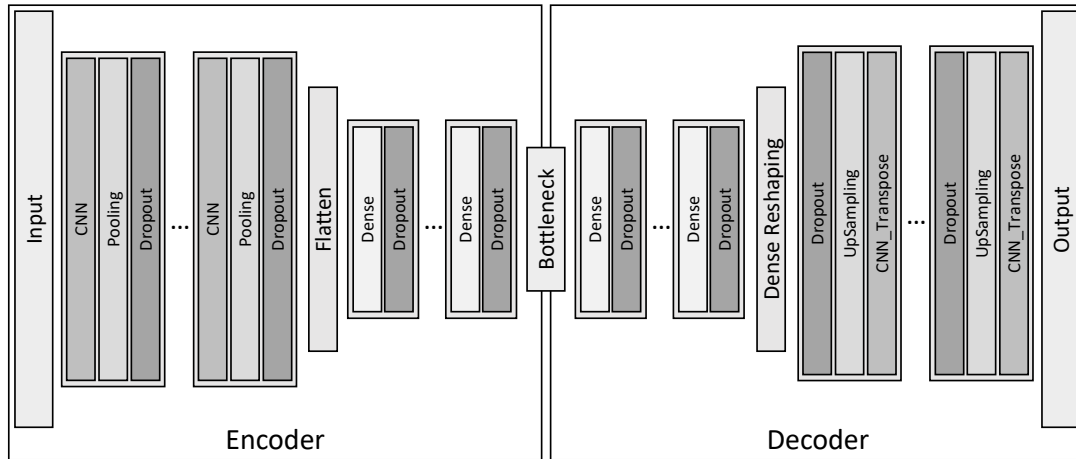
Figure 6: The Building Blocks of an Autoencoder: An Abstract Architecture. The AE's architecture comprises diverse blocks, each possessing unique internal attributes and dimensions. As a result, the encoder and decoder are constructed separately, deviating from the conventional symmetrical autoencoder. These blocks encompass convolutional layers and their associated operations alongside a dense block that integrates dense and dropout layers.

- Each convolutional layer is optimized with a specific number of **filters** in its operation, namely 16, 32, 64, and 128. Furthermore, the **kernel size** is tuned to either 8, 16, or 32. While it is possible to consider additional values for these parameters, doing so would increase the search space for the tuner.
- The **dropout layer** is optional for each CNN and DNN block. Possible **dropout rates** range is 0.1, 0.2, 0.3, 0.4, and 0.5.
- **Max pooling** is another optional layer in the CNN-Block but with a fixed pooling size of two.
- The range of **neurons** in a dense layer is 32, 64, 128, and 256, respectively.
- The **activation function** chosen for each layer in the autoencoder remains consistent, namely, the ReLu, Tanh, Sigmoid, or Softmax function. However, only the output layer of the decoder is individually tailored to these four functions.

C. Latent Space Plot

The encoder's output projection is shown in Figure 7. This figure displays the latent variables on a two-dimensional scale for easier interpretation. Each point on the plot corresponds to a mapped instance, representing a complete time series from the test dataset. The colors indicate the categorization by DBSCAN in the latent space: red points are outliers, orange points represent instances with manually detected deviations yet considered OK, and green points indicate cases with no apparent deviations, also classified as OK.
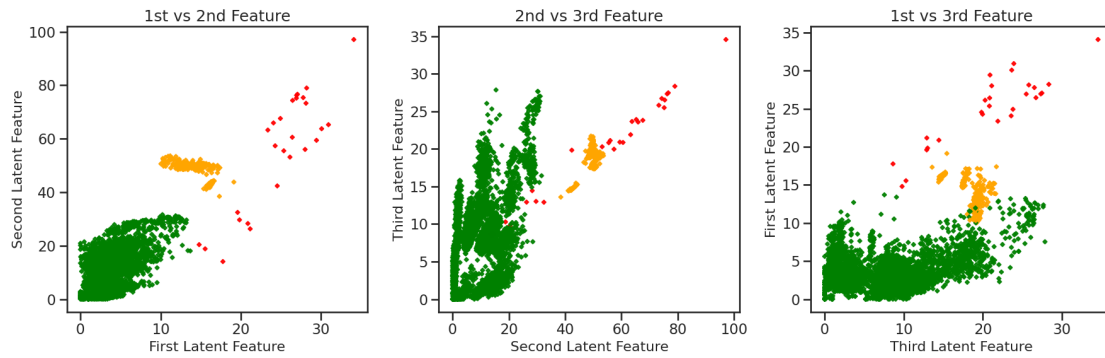
Figure 7: Latent Space Visualization. Two-dimensional latent space representation of the autoencoder's features. Green and orange points represent instances assigned to two distinct clusters, while red points are identified as outliers.

D. Exemplary Non-Outlier Time Series

Figure 8 displays an exemplary time series classified as a non-outlier alongside its reconstruction by the autoencoder. The image demonstrates that the AE can meaningfully reconstruct the input time series. Additionally, the pattern of this time series is typical for an instance without apparent errors in this dataset.
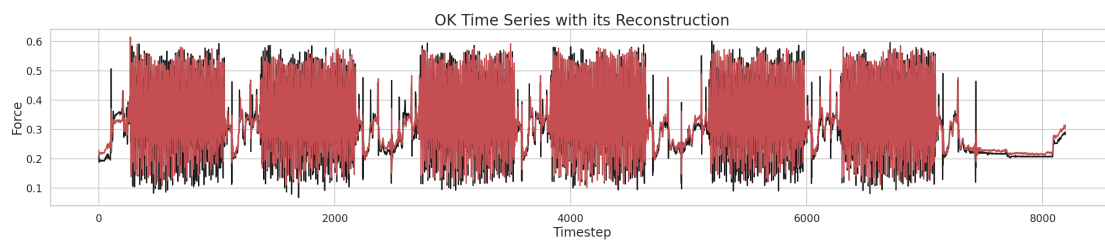

Figure 8: Time Series Reconstruction. The figure illustrates a time series, depicted in black, classified as OK. The corresponding reconstruction through the autoencoder is shown in red.