A Transformer-Based Framework for Anomaly Detection in Multivariate Time Series

Fabian Folger*, Murad Hachani†, Philipp Fuxen†, Julian Graf†,
Sebastian Fischer†, Rudolf Hackenberg†
Department of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Deutschland
* e-mail: fabian1.folger@st.oth-regensburg.de,
† e-mail: {murad.hachani | philipp.fuxen | julian.graf |
sebastian.fischer | rudolf.hackenberg}@oth-regensburg.de

***Abstract*—This paper introduces a comprehensive Transformer-based architecture for anomaly detection in multivariate time series. Using self-attention, the framework efficiently processes high-dimensional sensor data without extensive feature engineering, enabling early detection of unusual patterns to prevent critical system failures. In a subsequent laboratory setup, the framework will be applied using fuzzing techniques to induce anomalies in an Electronic Control Unit, while monitoring side channels, such as temperature, voltage, and Controller Area Network messages. The overall structure of the architecture, as well as the necessary preprocessing steps, such as temporal aggregation and classification up to the optimization of the hyperparameters of the model, are presented. The evaluation of the model architecture with the postulated restrictions shows that the model handles anomaly scenarios in the dataset robustly. It is necessary to evaluate the extent to which the model can be used in practical applications in areas, such as cloud environments or the industrial Internet of Things. Overall, the results highlight the potential of Transformer models for the automated and reliable monitoring of complex time series data for deviations.**

***Keywords-AI*; *Transformer*; *Time Series*; *Anomaly Detection*; *ECU*; *Temporal Aggregation*.**

I. INTRODUCTION

Transformer architectures have seen a surge in popularity recently, largely driven by the success of Large Language Models (LLMs) like ChatGPT, Gemini, and Claude. Initially focused on natural language processing tasks, these models have demonstrated that the underlying self-attention mechanisms can be beneficial in other domains as well. This trend is supported by the rapid increase in computing power in cloud and GPU environments, which now makes it possible to train and use models with a large number of parameters quickly and reliably.

A prime example is NVIDIA's recent presentation at CES, where new graphics cards and "DLSS 4" were introduced [1]. These products utilize transformer-based components to generate high-resolution pixels and entire frames, replacing the previously dominant Convolutional Neural Network (CNN) architectures with transformers that excel in parallel, context-sensitive processing.

Transformers are also becoming increasingly relevant for time series analysis, particularly when handling complex or multivariate sensor data. Their ability to capture long-range dependencies within signals is especially advantageous for anomaly detection—a critical need in industrial and Internet of Things (IoT) applications, such as machine, sensor, or network monitoring. Traditional methods like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) models, or CNNs often struggle with modeling long-term dependencies, making the self-attention mechanism of transformers a powerful alternative [2].

This work develops and evaluates a specialized transformer approach for anomaly detection on multivariate, labeled sensor data. The goal is to create a fully automated framework that can be applied to a variety of multivariate datasets, demonstrating how transformer models can detect rare abnormal states and outperform classical methods, such as LSTM autoencoders and Isolation Forests. Due to its ability to detect anomalies in complex sensor environments, this approach is particularly suitable for safety-critical applications, including its potential use in fuzz testing scenarios. The model is designed to support the automatic detection of anomalies in different domains; its effectiveness in fuzz testing environments will be evaluated in future work.

The paper is organized into seven sections. The Introduction provides an overview of the research topic and objectives. The Section II examines relevant approaches and previous studies. The Section III outlines the data sources and preprocessing steps. The Section IV details the model's design. The Section V reviews and interprets the results. The Section VI compares the model presented with other common methods in this domain. Finally, the Section VII summarizes key insights, highlights potential applications, and suggests directions for future research.

II. RELATED WORK

The detection of anomalies in technical systems is a comprehensive field of research that has gained considerable importance in recent years due to advances in the field of machine learning and, in particular, the use of neural networks. Various methods are employed, differing depending on the domain and data structure. This section presents related work from the fields of fuzz testing, transformer-based anomaly detection on time series data, and a platform for monitoring Electronic Control Units (ECUs) using side-channel analysis, which together provide a background for our research.

M. Böhme et al. have analyzed the challenges and opportunities of fuzzing and emphasize the problem of "human-in-the-loop", where test auditors have to invest considerable effort in analyzing the fuzzing results. Their research focuses on automating these processes through Artificial Intelligence (AI) [3]. L. McDonald et al. classify fuzzing methods and investigate hybrid approaches that combine static analysis, runtime error detection, and machine learning. They identify side-channel fuzzing as a promising extension for black-box fuzzing in the field of embedded [4]. One of the main problems with black box fuzzing is the lack of direct insight into the internal processes of the system under test. One solution to this is side-channel fuzzing, in which physical side channels, such as power consumption, electromagnetic emissions, or temperature curves are analyzed [5] [6]. P. Sperl et al. show that power trace analyses are suitable for optimizing fuzzing processes by drawing conclusions about the internal processes of a system [5]. This method has been successfully applied to embedded systems to identify unexpected behavior and detect error states more efficiently.

Transformer models have achieved outstanding performance in many tasks in the field of natural language processing and computer vision [7]. Transformer models have also demonstrated outstanding results in the analysis and forecasting of time series data [8][9]. Furthermore, it is important to point out that there is no obvious need to use transformer models for long-term time series predictions, since even simple MLP models can outperform transformers, such as DLinear in the analysis of long-term time series predictions [10]. However, the results of transformers, such as PatchTST [11] already show significant improvements. Xu et al. [2] were able to prove that transformer models also offer advantages in the detection of time series anomalies, since temporal dependencies can be represented by the model, which leads to a high detection performance. Although there are different transformer architectures that have already been applied to time series-based data with different focused objectives, such as lightweight [12] [13] or cross-block connectivity [14] or adaptive computation time [15], it is still a future task to adapt and evaluate different architectural approaches to time series data [7]. However, Zerveas et al. [16] have shown that there is little work on pre-trained transformers for time series data and the existing studies focus mainly on the classification of time series data. In various system architectures and frameworks that rely on the combined analysis of static and dynamic AI-based methods, such as [17], research into modern approaches to processing and anomaly detection of time series to improve the recognition rates of these systems and frameworks are important.

The monitoring and anomaly detection used in this work is based on the platform of Fuxen et al. [18] who have developed a system for monitoring ECUs using side-channel analysis for fuzz testing in future mobility systems. Based on this platform, our transformer model was developed and will be evaluated in a fuzz testing scenario as part of future work. The combination of monitoring, fuzzing and AI-supported analysis offers an innovative approach to safeguarding safety-critical systems and uses the findings of Fuxen et al. as a foundation.

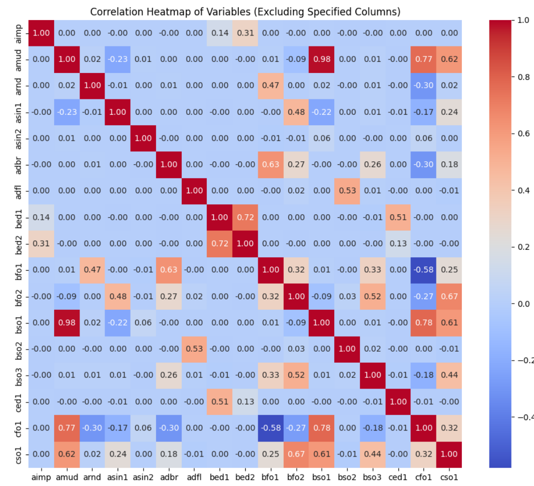

Figure 1. Correlation Heatmap

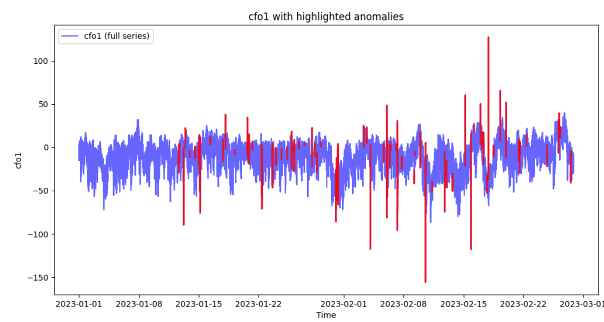

Figure 2. Time series variable cfo1 plotted with anomalies highlighted in red

III. Dataset

The primary criterion in selecting the dataset was its applicability to real-world scenarios, particularly for deploying the AI on real-time measurements in a laboratory environment. The Controlled Anomaly Time Series (CATS) dataset [19] from Solenix Engineering GmbH was chosen for its rich features that enable a realistic simulation of complex dynamic systems. This dataset comprises synthetically generated multivariate time series with 17 variables, including control commands, environmental influences, and sensor data, such as temperature and voltage.

At the outset of the data analysis, a correlation matrix was created in the form of a heatmap. This heatmap shown in Figure 1, depicts the complete dataset and reveals strong interdependencies among several variables. The color coding indicates negative correlations in blue and positive correlations in red—both types being crucial for the transformer when dealing with multivariate datasets. It is particularly noticeable that the characteristics *bso1* and *cfo1* with 0.98, and 0.77 show a relatively strong correlation with *amud*.

Figure 2 presents an example of injected anomalies in the variable CFO1, with anomalies highlighted in red. The plot was generated prior to data cleaning and scaling, showing raw measurement values on the Y-axis and the temporal progression

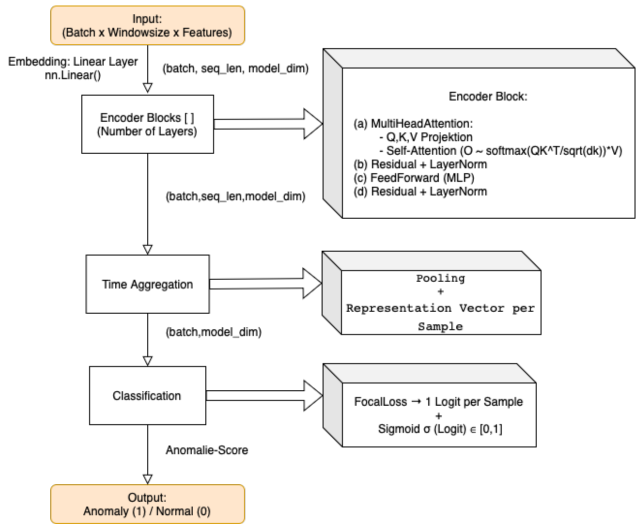

Figure 3. Transformer Architecture

(dates) on the X-axis, as the data were collected over several days.

A critical consideration for this dataset is the "root cause" of the anomalies, which initially manifests in other variables. Therefore, the dataset must not be truncated or abstracted, as doing so would eliminate these dependencies and prevent the transformer from learning the inherent patterns present in the original variables.

Notably, the dataset contains 200 carefully injected and annotated anomalies, which cover both obvious and context-dependent cases, making it an excellent benchmark for training the transformer-based anomaly detection system. With a resolution of 1 Hz over 5 million timestamps, the dataset offers data to learn normal system behavior and to develop robust anomaly detection models. An accompanying metadata file lists the time intervals of the anomalies, facilitating the removal of these segments during data preparation to expand the training set of normal data.

This dataset mirrors the physical measurement parameters expected in the laboratory. By applying the transformer model—known for its ability to capture both local and global temporal dependencies and to model complex multivariate correlations—to this synthetic data, the anomaly detection system can be evaluated and optimized under conditions that closely resemble practical scenarios.

IV. Architecture

In the following section, we provide an in-depth description of the implemented Transformer architecture. This section covers the core structure, the underlying mathematical principles. Figure 3 illustrates the overall Transformer architecture, which is designed to classify time series segments based on their anomaly probability. The architecture is structured into the following steps:

1) Input Layer: The initial processing stage for incoming time series segments.
2) Linear Embedding (Embedding-Layer): Transforms raw sensor data into a higher-dimensional representation.
3) Batch Normalization: Enhances training stability by normalizing feature distributions.
4) Transformer Encoder: Implements multi-head self-attention and feedforward networks to capture temporal dependencies.
5) Global Aggregation (Mean over Time): Condenses the time-dependent representations into a single feature vector per segment.
6) Classification Head (Linear Output): Projects the aggregated features into a scalar output for anomaly detection.

A. Input Layer and Batch Normalization

The model begins by processing time series segments, each with dimensions [BatchSize, WindowSize, Features]. A linear embedding layer transforms raw sensor data (17 features) into a higher-dimensional space (e.g., 128 dimensions) using the formula:

$$y = xA^T + b \tag{1}$$

where x is the input, A represents the weight matrix (initialized uniformly with

$$k = \frac{1}{\text{in\_features}} \tag{2}$$

and b the bias. This step is crucial in mapping the raw input data into a format suitable for the self-attention mechanism.

Batch normalization is applied to maintain a stable distribution of features across each sliding window. This normalization improves the model's robustness during training.

B. Attention and Encoder Layer

In our model, we adopt the vanilla Transformer architecture as described in Attention is All You Need [20]. The attention mechanism begins by projecting the input into three distinct representations—queries, keys, and values—via learned linear transformations. These projections are then used in a scaled dot-product attention computation, where the dot product of queries and keys is scaled by the inverse square root of the key dimensionality to ensure numerical stability. The resulting attention weights are applied to the values, allowing the model to focus on relevant parts of the input.

Building upon this, the encoder layer integrates multi-head self-attention with a position-wise feed-forward network. Each encoder block applies residual connections and layer normalization both after the multi-head self-attention and the feed-forward network, which enhances gradient flow and stabilizes training. This combination of attention and encoder components forms the core of the Transformer model, enabling it to capture complex dependencies in sequential data.

C. Temporal Aggregation

After processing the sequence through the Transformer encoder layers, a temporal aggregation step is applied to condense the time-dependent representations into a single vector per input segment. This is achieved using an element-wise mean pooling operation across the time dimension:

$$\mathbf{z} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t,$$

where $\mathbf{x}_t \in \mathbb{R}^{d_{\text{model}}}$ is the output for each time step t and z represents the aggregated feature vector. Although alternatives, such as max pooling or token-based representations (e.g., using a `[CLS]` token) exist, average pooling is chosen here for its simplicity and its ability to equally represent all time steps [21][22].

D. Classification

The final stage of the architecture is the classification head. The aggregated vector z is passed through a linear layer to produce a scalar output:

$$o = \text{Linear}(\mathbf{z}) \in \mathbb{R}.$$

A sigmoid activation is then applied to convert the scalar into a probability score between 0 (normal) and 1 (anomalous) [21][22]:

$$\sigma(\text{Logit}) = \frac{1}{1+\exp(-\text{Logit})}$$

A threshold value determines the final binary classification. For handling class imbalances typical in anomaly detection, a Focal Loss is used instead of standard binary cross entropy [23]. The Focal Loss formula is:

$$\text{FL}(\hat{y}, y) = \alpha\left(1-\hat{y}\right)^{\gamma}\text{BCE}(\hat{y}, y),$$

this is set to emphasize hard-to-classify examples.

Anomaly detection typically involves highly imbalanced data, where anomalous events are extremely rare compared to normal instances. Traditional loss functions like Binary Cross Entropy (BCE) treat all samples equally, often causing the model to be biased toward the majority class and overlook the few but critical anomalies. Focal Loss addresses this challenge by dynamically down-weighting the loss contribution of well-classified normal samples and up-weighting the misclassified or harder-to-classify anomalous examples. In practice, this means that the model is encouraged to learn more from the sparse anomaly examples, enhancing its sensitivity and overall detection performance in an imbalanced dataset [23].

E. Training

In preparation for training, suitable hyperparameters for the Transformer model are determined using Optuna [24]. This process follows a semi-supervised approach: while the model is mainly trained on normal (non-anomalous) data, the validation set contains a few anomalies. This setup enables the optimization process to favor parameter combinations that effectively detect anomalies, optimizing for metrics, such as the F1 score.

Once Optuna identifies the best hyperparameters, the model is retrained on normal data in an unsupervised manner—meaning it does not explicitly see any anomaly examples during training. The final evaluation, however, is performed on a test set containing both real and synthetically generated anomalies, thereby assessing the model's true capability to detect unusual patterns.

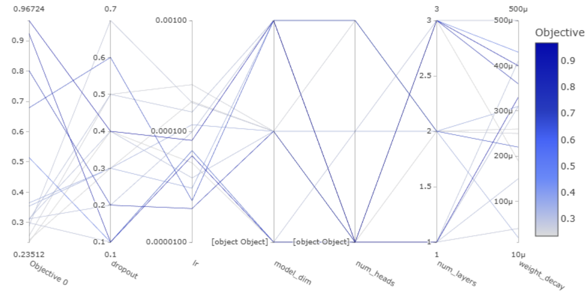

Figure 4. Evaluation of Hyperparameters using Optuna

The hyperparameter tuning is managed via an Optuna Optimizer class (see Figure 4), which integrates several DataLoader instances for training, validation, and test data (both normal and anomalous). During each trial in the process, hyperparameters (e.g., model dimension, number of attention heads, encoder layers, dropout, learning rate, and weight decay) are sampled from predefined ranges.

V. Evaluation

To assess the model's performance, we applied standard binary classification metrics—such as the confusion matrix, F1-score, and ROC-AUC—among others. In this section, we analyze the results on the CATS dataset using these metrics, and we detail the hyperparameter configuration of the best performing model as identified through OPTUNA [25][26].

TABLE I
Model Results

Transformer Metrics	Values
Optimal threshold	0.0117
ROC-AUC	0.9993
F1-Score	0.9717
Precision	0.9585
Recall	0.9853
Accuracy	0.9921
Anomalies detected (all Labels)	11731/83558 (14.04%)
Anomalies detected (Anomaly-Labels)	11244/11413 (98.52%)

Table I summarizes the overall performance of our anomaly detection model. The high ROC-AUC of 0.9993 and F1-Score of 0.9717 indicate that the model is effective in discriminating between normal and anomalous instances in the CATS dataset while being trained with an unsupervised method. The model achieves a precision of 0.9585 and a recall of 0.9853, which reflects its balanced capability to correctly identify anomalies while minimizing false positives. Overall the accuracy is reported at 0.9921, and the confusion matrix confirms that 98.52% of the true anomaly labels are correctly detected. These metrics underscore the robustness of our approach in handling complex multivariate time series data and highlight its potential for reliable anomaly detection in practical applications. The confusion matrix confirms these results, with the model detecting 98.52% of true anomalies.

Key hyperparameters for the CATS dataset and optimized via Optuna include:

- **Dropout:** 0.2
- **Learning Rate:** 2.0075e-05
- **Model Dimension:** 128
- **Attention Heads:** 8
- **Encoder Layers:** 3
- **Weight Decay:** 0.00036
- **Batch Size:** 128

VI. Comparison With Other Architectures

In this section, we provide a preliminary comparative analysis of our Transformer-based anomaly detection model with the established methods evaluated in the publication *Anomaly Detection in Time Series: A Comprehensive Evaluation* [26]. It should be noted that our evaluation is currently limited to the CATS dataset, which was chosen due to its close resemblance to our laboratory setup. This selection enables us to investigate the model's performance under controlled conditions that reflect our specific experimental environment.

While a direct comparison is inherently challenging due to the use of different datasets across studies, we have compared our model's performance with established architectures by benchmarking key metrics (ROC-AUC, F1-Score, Precision, and Recall) against those reported in the literature. We acknowledge that such indirect comparisons have limitations, however, on the Timeeval website, the CATS dataset is already listed and integrated on GitHub [27]. This availability opens up opportunities for further evaluation, either by our team or by the broader research community. In future work, we plan to leverage this integration to perform additional assessments and comparisons. Furthermore, benchmarking our Transformer-based model using the datasets available on Timeeval could provide a standardized framework to rank and compare its performance against other state-of-the-art approaches in [26].

Custom aspects of our framework include:

- **Focal Loss with Self-Attention:** This combination, while less common in time series applications, effectively addresses class imbalance by emphasizing the misclassified anomaly cases.
- **Flexible Time Window Segmentation:** The model adapts its sequence length and optimizes the number of attention heads, allowing it to better capture diverse temporal characteristics.
- **Tailored Binary Anomaly Classification:** By focusing on binary labels (normal vs. anomalous) and leveraging a specialized loss function, our approach directly targets the detection of rare anomaly labels.

A key discussion point is the generalizability of the results. Although the dataset used in this study contains carefully embedded anomalies, it remains an open question whether these findings can be directly transferred to more complex, real-world scenarios. Future experiments with time series data from various domains—such as industrial processes or medical applications—are needed to fully assess the model's robustness under varying conditions.

Another important aspect is the sensitivity to preprocessing choices. For instance, the window size used in the sliding-window technique has a significant impact on the patterns that are detected based on our experiments. Similarly, strategies for handling missing data and the specifics of hyperparameter tuning (e.g., via Optuna) can greatly influence the reliability and speed of anomaly detection. Sensitivity analyses, such as systematically varying the window size or testing different missing-data methods, would help ensure that the model's performance is not overly dependent on narrow parameter settings.

Compared to traditional methods, the Transformer—especially in its encoder-only configuration—offers potential advantages in terms of flexibility and performance. While decoder layers might be beneficial for forecasting or reconstruction tasks, they also increase complexity and resource requirements without necessarily improving pure anomaly detection. In contrast, older methods like classical autoencoders or statistical approaches (e.g., Isolation Forest) can sometimes achieve similar results with less effort but often struggle to capture the complex, nonlinear dependencies in high-dimensional time series data.

VII. Conclusion and Future Work

The Transformer-based architecture presented in this work has proven to be a potential approach for anomaly detection in multivariate time series. By leveraging self-attention mechanisms, the model was able to capture relationships in the dataset and accurately predict potential anomalies with minimal feature engineering. The flexible data pipeline—including missing-value handling, scaling, and segmentation—enables rapid adaptation to new datasets and application scenarios.

Optimized hyperparameter optimization has shown that a systematic, semi-supervised approach can identify optimal settings that effectively support the final unsupervised training. However, the validity of these results is highly dependent on the quality and representativeness of the underlying data. Therefore, it is necessary to perform future evaluations on different real data sets from different domains using time series data to finally confirm the generalizability and robustness of the model.

Overall, the experiments highlight the potential of transformer architectures for anomaly detection and provide valuable insights for future research.

In practice, the Transformer-based approach offers the advantage of capturing high-dimensional relationships between sensor variables. This capability enables the reliable identification of unusual patterns even in noisy environments or under changing data distributions. It can be concluded that the demonstrated method has some relevance for industrial or safety-critical scenarios in which multi-sensor data must be continuously monitored for deviations. However, further development is required to provide a productive model for this.

Comparing models across heterogeneous datasets remains challenging. Our preliminary benchmarking using key metrics underscores the potential of our approach. The integration of the CATS dataset into Timeeval offers a promising avenue for future standardized evaluations of our Transformer-based model alongside other state-of-the-art techniques [26].

Moreover, Transformer-based anomaly detection is finding increasing application in other fields, such as medicine, where it can identify abnormal patterns in complex biosignals. In our laboratory setup, an ECU combined with a fuzzer is used to deliberately induce anomalous states on the Controller Area Network (CAN) bus while simultaneously capturing side-channel data, such as temperature and voltage. This integrated approach not only delivers a comprehensive view of the system state but also provides feedback to progressively enhance the fuzzing algorithm [18].

The results of the presented methodology for time-series-based anomaly detection confirm the effectiveness of the developed model within the limitations. This provides a solid basis for further investigations and evaluations in laboratory environments and beyond. The flexible exploration of different use cases within the research group with regard to automotive security and IoT security is a focused goal.

References

[1] NVIDIA, "Dlss 4: Multi-frame generation and ai innovations", Accessed: 2025.03.19, NVIDIA Corporation, 2024, [Online]. Available: https://www.nvidia.com/en-us/geforce/news/dlss4-multi-frame-generation-ai-innovations/.

[2] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy", *CoRR*, vol. abs/2110.02642, 2021. arXiv: 2110.02642.

[3] M. Böhme, C. Cadar, and A. Roychoudhury, "Fuzzing: Challenges and reflections", *IEEE Software*, vol. 38, no. 3, pp. 79–86, 2021.

[4] L. McDonald, M. I. U. Haq, and A. Barkworth, "Survey of software fuzzing techniques", Dec. 2021.

[5] P. Sperl and K. Böttinger, "Side-channel aware fuzzing", in *Computer Security–ESORICS 2019: European Symposium on Research in Computer Security*, Springer, vol. 24, Sep. 2019, pp. 259–278, ISBN: 978-3-030-29958-3. DOI: 10.1007/978-3-030-29959-0_13.

[6] M. Muench, J. Stijohann, F. Kargl, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: Challenges in fuzzing embedded devices", *NDSS Symposium 2018*, 2018. DOI: 10.14722/ndss.2018.23176.

[7] Q. Wen *et al.*, *Transformers in time series: A survey*, 2023. arXiv: 2202.07125.

[8] S. Li *et al.*, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting", *Advances in neural information processing systems*, vol. 32, 2019.

[9] T. Zhou *et al.*, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting", in *International conference on machine learning*, PMLR, 2022, pp. 27 268–27 286.

[10] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, 2023, pp. 11 121–11 128.

[11] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, *A time series is worth 64 words: Long-term forecasting with transformers*, 2023. arXiv: 2211.14730 [cs.LG].

[12] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention", *CoRR*, vol. abs/2004.11886, 2020. arXiv: 2004.11886.

[13] S. Mehta, M. Ghazvininejad, S. Iyer, L. Zettlemoyer, and H. Hajishirzi, "Delight: Very deep and light-weight transformer", *CoRR*, vol. abs/2008.00623, 2020. arXiv: 2008.00623.

[14] A. Bapna, M. X. Chen, O. Firat, Y. Cao, and Y. Wu, "Training deeper neural machine translation models with transparent attention", *CoRR*, vol. abs/1808.07561, 2018. arXiv: 1808.07561.

[15] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal transformers", *CoRR*, vol. abs/1807.03819, 2018. arXiv: 1807.03819.

[16] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning", in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2114–2124.

[17] J. Graf, K. Neubauer, S. Fischer, and R. Hackenberg, "Architecture of an intelligent intrusion detection system for smart home", in *2020 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops)*, IEEE, 2020, pp. 1–6.

[18] P. Fuxen, M. Hachani, J. Schmidt, P. Zaumseil, and R. Hackenberg, "Side channel monitoring for fuzz testing of future mobility systems", *CLOUD COMPUTING 2023*, 2023.

[19] Patrick Fleith, *Controlled anomalies time series (CATS) dataset*, version 2, Sep. 14, 2023. DOI: doi.org/10.5281/zenodo.7646896.

[20] A. Vaswani *et al.*, "Attention is all you need", *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762.

[21] S. Song, N.-M. Cheung, V. Chandrasekhar, and B. Mandal, "Deep adaptive temporal pooling for activity recognition", in *Proceedings of the 26th ACM international conference on Multimedia*, Oct. 15, 2018, pp. 1829–1837. DOI: 10.1145/3240508.3240713. arXiv: 1808.07272[cs].

[22] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, *A transformer-based framework for multivariate time series representation learning*, Dec. 8, 2020. DOI: 10.48550/arXiv.2010.02803. arXiv: 2010.02803[cs].

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, Feb. 7, 2018. DOI: 10.48550/arXiv.1708.02002. arXiv: 1708.02002[cs].

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework", in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Jul. 25, 2019, pp. 2623–2631, ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330701.

[25] P. Wenig, S. Schmidl, and T. Papenbrock, "TimeEval: A benchmarking toolkit for time series anomaly detection algorithms", *Proc. VLDB Endow.*, vol. 15, no. 12, pp. 3678–3681, Aug. 1, 2022, ISSN: 2150-8097. DOI: 10.14778/3554821.3554873.

[26] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation", *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, May 2022, ISSN: 2150-8097. DOI: 10.14778/3538598.3538602.

[27] T. Project, "Timeeval - datasets overview", Accessed: 2025.03.19, 2022, [Online]. Available: https://timeeval.github.io/evaluation-paper/notebooks/Datasets.html.