Towards Enhanced IoT Security: Advanced Anomaly Detection using Transformer Models

Natalia Sánchez[1], Albert Calvo[1], Santiago Escuder[1], Josep Escrig[1], Jordi Domenech[1], Nil Ortiz[1], and Saber Mhiri[1]

[1]i2CAT Foundation, Barcelona, Spain

***Abstract*—The proliferation of Internet of Things (IoT) devices has significantly increased data traffic, necessitating robust security measures to protect against latent threats. Traditional anomaly detection methods often struggle to keep pace with the dynamic and diverse nature of IoT environments, particularly in use cases with limited accessible training data. This necessitates adaptive and efficient solutions. In this work in progress, we propose the use of fine-tuned Transformer models for anomaly detection in an IoT traffic use-case. These techniques allow adjusting a pre-trained model to a new domain where scarce training data is accessible. Specifically, we explore the usage of fine-tuning Transformer architectures using the CIC IoMT 2024 dataset and evaluate it with the Aposemat IoT-23 dataset. Compared to traditional machine learning techniques, our proposed approach demonstrates promising performance improvements, bridging the gap in developing novel Transformer-based architectures capable of providing supervised fraud detection capabilities, even with highly limited datasets for training.**

***Index Terms*—Cybersecurity, Machine Learning, Anomaly Detection, IoT (Internet of Things), Transformer Models,**

I. Introduction

The large usage of IoT devices has led to an exponential increase in data traffic, making anomaly detection a critical aspect of cybersecurity. This massive growth of data can be useful, but it also hides potential dangers. Undetected anomalies in this traffic can have severe consequences, ranging from data breaches exposing sensitive information to physical damage caused by malfunctioning devices. Infiltration by malicious actors can lead to privacy violations or even compromise critical infrastructure. Traditional anomaly detection methods, while valuable, often struggle to keep pace with the dynamic and diverse nature of IoT environments. The latent research in Transformer-based architectures and especially in fine-tuning processes enable efficient, transfer learning, context-aware models, which can be applied to a wide range of use cases. Using fine-tuning techniques presents a promising solution due to its ability to be trained on large datasets and adapted to the particularities of new use cases with limited data.

The present work in progress involves the application of fine-tuned Transformer models for anomaly detection in the traffic data of IoT devices. The proposed approach uses transfer learning, achieving preliminary satisfactory results on public datasets, while also enabling adaptation to other datasets with different IoT scenarios. We believe that the proposed work is a promising step towards better anomaly detection and will advance the security posture of IoT systems, enabling the development of more sophisticated threat mitigation strategies.

II. Motivation

Anomaly detection in real-world scenarios can be challenging due to the myriad of different attacks. Additionally, public data available for benchmarking proposed data-driven approaches is limited. In this section, we present two public datasets, CiCIoMT-24 and IoT-23, which will be used for benchmarking our proposed experiments. We also provide preliminary ground truth results, involving training several models on the CiCIoMT-24 dataset and performing inference using the IoT-23 dataset. These datasets are among the most recent publicly available collections of network IoT anomaly data.

- The **CIC IoMT 2024** dataset comprises a collection of network traffic, including both normal operations and malicious activities targeted at IoT medical devices, also referred to as IoMT [3]. This dataset integrates 40 IoMT devices, consisting of 25 real devices and 15 simulated units, across various communication protocols, including Wi-Fi, MQTT, and Bluetooth. The attacks captured in this dataset are diverse, covering five distinct categories: Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance (Recon), MQTT-specific attacks, and Spoofing. This categorisation helps simulate a realistic array of threats common in the IoMT landscape.
- The **Aposemat IoT-23** dataset also contains both malicious and benign network traffic, providing a realistic perspective on potential cybersecurity threats of domestic IoT devices [5]. The dataset comprises 23 detailed scenarios, with 20 captures related to network traffic generated by malware executed on a Raspberry Pi. The malware executed comprises activities such as DDoS, C&C, and Mirai attacks. The remaining three scenarios represent normal behavior from three different IoT devices.

Distinct models were initially trained in the CICIoMT2024 dataset to provide a baseline, including supervised classification algorithms such as K-Nearest Neighbors (KNN), Random Forest (RF), and XGBoost. Additionally, two deep learning based models were implemented: an LSTM-based architecture and a Transformer-based architecture. To ensure fairness in the non-deep learning models, the dataset was balanced so that each label had a more equal representation. This balancing was achieved using upsampling and downsampling methods. This preprocessing step was only applied to the KNN and RF models, as the XGBoost model automatically adjusts the weights based on the number of samples in each class. In Table I

is included a benchmarking of the classification performance metrics for the trained models including the performance metrics of Accuracy, Precision and Recall and F1-Score, which computes the harmonic mean between precision and recall. In the stated experiments, a binary classification task is executed to distinguish between benign and malicious traffic data. The CICIoMT2024 dataset contains 7,500 samples, of which over 5,200 are malicious. For the multi-class classification, these malicious samples are categorised into 2,154 samples for DDoS attacks, 2,128 for DoS attacks, 354 for Recon attacks, 331 for MQTT attacks, and 310 for Spoofing attacks.

TABLE I
CLASSIFICATION PERFORMANCE METRICS IN CICIOMT2024

Model	Type	Acc.	Prec.	Rec.	F1
KNN		0.85	0.85	0.83	0.84
RF		0.86	0.86	0.86	0.86
XGBoost		0.87	0.89	0.86	0.87
LSTM	Binary	0.68	0.59	0.64	0.60
LSTM + T2V		0.70	0.69	0.68	0.69
S. Transformer		0.78	0.79	0.78	0.78
T. Transformer		0.80	0.80	0.80	0.79
KNN		0.71	0.47	0.35	0.38
RF		0.73	0.56	0.49	0.52
XGBoost		0.78	0.68	0.50	0.56
LSTM	Multi-class	0.69	0.57	0.32	0.39
LSTM + T2V		0.74	0.61	0.35	0.47
S. Transformer		0.66	0.61	0.67	0.60
T. Transformer		0.73	0.68	0.73	0.68

Analysing the Table I provides a comparison of the baseline models with a basic transformer architecture, including both its versions, with (S. Transformer) and without the time encodings (T. Transformer). The architecture of the Transformer includes an encoder with multiple self-attention layers and position-wise fully connected feed-forward networks, enhanced with positional encoding to incorporate sequence order information. The Transformer is trained to reconstruct normal data sequences; deviations from these reconstructions are analysed to detect anomalies. The results indicate significant variations in performance across different models and classification types. For the binary task, where is classified by benign or malign traffic data, the XGBoost model shows the highest F1-score at 0.87, closely followed by the Transformer-based architecture with an F1-score of 0.785. However, for the LSTM-based models, the performance is notably lower compared to other models with scores ranging from 0.60 to 0.69.

In contrast, for the multi-class classification task, where different attack types are identified, the performance drops across all models compared to the binary task. XGBoost still maintains the highest F1-score among all statistical models at 0.56, with the Transformer-based architecture without time encodings benign slightly better at 0.60. In this task, the Transformer with time encoding has the highest score among all models at 0.67. However, there is a considerable decrease in precision, recall, and F1 scores across all models for the multi-class classification task compared to the binary task.

III. RELATED WORK

A. Anomaly detection Transformer

Transformers have been increasingly applied to time series tasks, including anomaly detection, due to their ability to handle complex dependencies and large datasets efficiently. Several Transformer-based models have emerged, each with unique strengths for anomaly detection, key contributions and variants. As an example, Anomaly Transformer [17], focuses on capturing temporal dependencies and distinguishing normal patterns from anomalies in time series data. It combines attention mechanisms to effectively highlight anomalies in IoT log data by comparing the expected and actual patterns within the data sequences. Various adaptations of the Transformer architecture have been proposed to improve its effectiveness in anomaly detection. These include the incorporation of positional encodings focused on time series data [13], multi-head attention to capture diverse aspects of the data, and modifications at both the module and architectural levels to better accommodate the specific characteristics of IoT log data [15]. Studies have demonstrated the superiority of Transformers over traditional and other deep learning methods in terms of anomaly detection performance. For instance, empirical analysis have shown that Transformers achieve higher accuracy and lower false positive rates, making them a robust choice for monitoring and maintaining the security of IoT networks [15].

B. Fine-tuning methods

Fine-tuning is a specific form of transfer learning where the pre-trained model is adjusted to make it more suitable for a specific task. This is achieved by continuing the training process on a new dataset, allowing the model to adjust its weights to better accommodate the specifics of the target task. Fine-tuning is widely used to adapt models that have been trained on large, generic datasets to perform well on smaller, specific datasets. This method is particularly beneficial when the new dataset is small, as it helps in retaining the pre-learned features while adjusting to new data [18], [9].

Domain adaptation is a branch of transfer learning focused on adapting a model to work under different distribution conditions than those it was originally trained on. This is crucial for practical applications where data collected in real-world conditions often differ significantly from the data used during training [4].

- **Domain-Invariant Feature Learning**: The core idea is to align the source and target domains by creating a feature representation that is invariant across domains, meaning the features follow the same distribution regardless of the domain of the input data. This approach assumes the existence of such a feature representation and that the marginal label distributions between domains do not significantly differ.
- **Divergence-Based Methods**: These methods aim to minimise a divergence that measures the distance between the source and target distributions. Techniques include maximum mean discrepancy (MMD) [10], correlation

alignment (CORAL) [11], contrastive domain discrepancy (CCD) [6], and the Wasserstein metric [8], each with its way of measuring and minimising distributional discrepancies.

- **Adversarial Methods**: This category includes feature-level adversarial domain adaptation methods, where the alignment component often consists of a domain classifier. This classifier outputs whether the feature representation came from source or target data, and the system is trained in such a way that the classifier cannot accurately determine the domain of the feature representation, making the features domain-invariant. This approach makes use of adversarial training techniques similar to those used in Generative Adversarial Networks (GANs) to achieve domain adaptation [12].

After the literature review, it is decided to employ domain-invariant feature learning for our transfer learning approach for several reasons. Firstly, the task remains consistent across both the source and target domains. Secondly, the types of features are uniform across datasets. Finally, the labels can be adapted to accommodate the variations between domains. By focusing on aligning the features and classes across domains, we aim to ensure that our model can effectively generalise its learning from the source domain to the target domain, despite potential differences in data distributions.

IV. Proposed Methodology

This section presents an overview of the proposed data-driven workflow for experimentation. In Section subsection IV-A, we elaborate on the feature engineering process. Section subsection IV-B details the transformer architecture employed. Lastly, Section subsection IV-C discusses the various transfer learning techniques implemented.

A. Feature extraction and creation of time windows

The raw network data from the different datasets, compressed in pcap files, is pre-processed using Zeek software to obtain the network connection logs [1]. From these files, we process the network traffic data to extract and compute various statistical features. The pre-processed data is grouped by device, with each device identified by an IP address, and segmented into fixed 10-second time intervals. For each time sequence, a set of statistical features, as defined in [2], is computed, creating what we refer to as *samples*. This time interval is chosen to summarise the data effectively and capture enough information, as some attacks may span from a few seconds to several minutes. These samples can be grouped into *time windows* to further exploit temporal dependencies. During the creation of time windows, some intervals may have missing data. In such cases, we fill the missing intervals with zeros. If an entire time window lacks information or if the number of non-zero samples falls below a certain threshold, we discard that window.

B. Transformers architecture

The model employed in this work is a Transformer model, recognised as a state-of-the-art approach for processing sequential data [14].It employs a self-attention mechanism to weigh the importance of different parts of the input data, making it suitable for tasks where understanding the context and relationships within the data is crucial.

After experimenting with it, we decided to add a system for encoding the temporal information on the dataset, which we called "Time Transformer". This architecture, shown in Figure 1, combines the previous basic Transformer model with a time encoding to handle temporal data more effectively. This model uses the Time2Vec encoding to transform timestamps to a higher-dimensional space, using linear and sinusoidal transformations [7]. These transformations capture the patterns and trends in the timestamps and make a representation of the temporal information. These encoded values are then concatenated with the input features inside the transformer architecture which merge the feature and temporal information. This combination allows the model to learn the patterns of the temporal information as it is updated in each epoch.

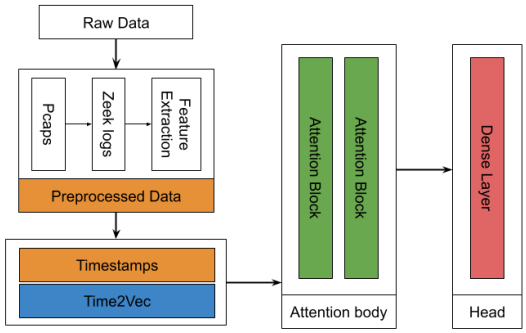

Fig. 1. Transformer Architecture

C. Transfer Learning

This section describes three different implemented domain-invariant fine-tuning techniques used to effectively apply transfer learning from models trained on the CIC IoMT 2024 dataset to the Aposemat IoT-23 dataset [16]:

- **New Output Layer**: This technique involves replacing the output layer, which is reinitialised while all other layers remain intact, and training continues from that point.
- **Layer Freezing**: The bottom layers are frozen, and the output layer is reinitialised. Training continues for a few epochs with the same learning rate.
- **Reduced Learning Rate**: This technique involves lowering the learning rate, which facilitates better adaptation with minimal performance changes.

V. Experimentation and Results

In this section, we present the performance metrics of various models applied to different datasets, comparing their effectiveness in both multi-class and binary classification tasks. To evaluate the performance of the models, the metrics considered include Accuracy, Precision, Recall, and F1 Score. The baseline model is an XGBoost model for the multi-class and binary tasks, this model takes into account the imbalance of

the classes in the dataset. Once the model is trained in the initial dataset (CIC IoMT2024), we can test it on an unseen dataset, in our case the IoT-23 dataset. To do so we adapted the different classes of the IoT-23 to the ones found in the CICIoMT dataset.

Each attack type in the IoT-23 dataset is mapped to a corresponding attack type in the CICIoMT2024 dataset based on the similarity of their behavior and the nature of the attack. This approach ensures that the model can work with the same labels across different datasets, thereby facilitating the evaluation of its performance. For instance, a Distributed Denial-of-Service (DDoS) attack in both datasets involves large-scale attempts to disrupt services, while "Spoofing" and "Hide & Seek" both involve deceptive techniques to bypass security measures. Similarly, mapping PortScan in the IoT-23 dataset to Recon in the CICIoMT2024 dataset is logical because PortScan is a type of reconnaissance attack where attackers scan a range of ports on a target system to discover open ports and the services running on them. Mapping complex and varied attacks into a simplified set of categories helps reduce the complexity of the multi-class classification task.

Due to the scarcity of samples for some attacks, as well as those for which we could not find equivalents among the previously encountered attacks, we decided to only select the threats of DDoS and PortScan. After the mapping, we end up with 20,500 benign samples and 6,362 malign samples, of which 5,328 are of the Recon attack and 1,034 are of the DDoS attack. If we make the inference of the processed IoT-23 dataset in the models trained with the CICIoMT2024 the results are subomptimal, as shown on Table II. This is expected, as the datasets differ significantly in the kind of devices used and the way of collecting the information of the attacks. To address this issue, we tested three techniques of fine-tuning described in subsection IV-C, and the outcomes were highly consistent across all of them. Even within a few epochs—such as at 5 epochs—the achieved results were notably high, with additional improvements occurring at a gradual pace. Given the similarity in performance among the fine-tuning methods, we decided to present a single row of metrics in the table, as they all yielded comparable values. This also emphasises the stability and effectiveness of the fine-tuning process across various techniques.

The experiments show that fine-tuning improves model performance, especially for datasets that exhibit class imbalances. When it comes to the binary classification tasks on unbalanced datasets, the baseline model performs well, but it is less successful in the multi-class task. Fine-tuning helps the Simple Transformer and Time Transformer models achieve high performance metrics, particularly in binary classification scenarios. These can also be attributed to the abundance of benign samples within the IoT-23 dataset compared to the relatively fewer instances of attacks, even after filtering out non-essential windows. With its persistent higher performance, the Transformer + T2V model is a promising method for binary and multi-class classification tasks.

TABLE II
CLASSIFICATION PERFORMANCE METRICS IN CICIOMT2024 WITH FINNE-TUNING AND VALIDATION USING THE APOSTEMAT IOT-23

Model	Type	Dataset	Acc.	Prec.	Rec.	F1
Baseline	Multi-class	CIC24	0.78	0.68	0.50	0.56
		IoT23	0.23	0.96	0.23	0.29
	Binary	CIC24	0.87	0.89	0.86	0.87
		IoT23	0.67	0.48	0.49	0.47
S. Transformer	Multi-class	CIC24	0.66	0.60	0.66	0.60
		IoT23	0.26	0.27	0.26	0.24
		IoT23+FT	0.90	0.91	0.90	0.88
	Binary	CIC24	0.78	0.79	0.78	0.77
		IoT23	0.44	0.72	0.44	0.31
		IoT23+FT	0.96	0.96	0.96	0.96
T. Transformer	Multi-class	CIC24	0.73	0.67	0.73	0.67
		IoT23	0.27	0.28	0.27	0.24
		IoT23+FT	0.96	0.96	0.96	0.95
	Binary	CIC24	0.79	0.80	0.79	0.78
		IoT23	0.52	0.63	0.52	0.49
		IoT23+FT	0.96	0.96	0.96	0.96

VI. CONCLUSIONS AND FUTURE WORK

In our ongoing research, we analysed the applicability of fine-tuning methodologies using two IoT traffic datasets that encompass similar threat and use-case scenarios. The experimental results demonstrate the promise of our approach, with significant improvements in performance for both binary and multi-class classification tasks. Our future work will focus on conducting more extensive experimentation with the transformer-based model, testing the proposed framework on network data collected from a physical use-case, and extending transformer-based models for better contextualisation to address the particularities of anomaly detection in the cybersecurity domain.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101019645, SECANT project, and from the CDTI CERVERA research and innovation program, supported under Grant Agreement CER-20231019 CICERO. The content of this article does not reflect the official opinion of the European Union, the CDTI, or any other institution. Responsibility for the information and views expressed therein lies entirely with the authors.

REFERENCES

[1] Paxson V Bro. A system for detecting network intruders in real-time. In *Proc. 7th USENIX security symposium*, 1998.

[2] Albert Calvo, Santiago Escuder, Josep Escrig, Marta Arias, Nil Ortiz, and Jordi Guijarro. A data-driven approach for risk exposure analysis in enterprise security. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9. IEEE, 2023.

[3] Sajjad Dadkhah, Euclides Carlos Pinto Neto, Raphael Ferreira, Reginald Chukwuka Molokwu, Somayeh Sadeghi, and Ali Ghorbani. Ciciomt2024: Attack vectors in healthcare devices-a multi-protocol dataset for assessing iomt device security. 2 2024.

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, 2015.

[5] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. Iot-23: A labeled dataset with malicious and benign iot network traffic (version 1.0.0), 2020.

[6] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019.

[7] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[8] Jian Shen, Yezhou Qu, Weiran Zhang, and Yinghui Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI Conference on Artificial Intelligence*, pages 4058–4065, 2018.

[9] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Zeju Xu, Isabel Nogues, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[10] Alexander J Smola, A Gretton, and K Borgwardt. Maximum mean discrepancy. In *13th international conference, ICONIP*, pages 3–6, 2006.

[11] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. *Domain adaptation in computer vision applications*, pages 153–171, 2017.

[12] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[13] Waseem Ullah, Fath U Min Ullah, Zulfiqar Ahmad Khan, and Sung Wook Baik. Sequential attention mechanism for weakly supervised video anomaly detection. *Expert Systems with Applications*, 230:120599, 2023.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. 2 2022.

[16] Xueli Xiao, Thosini Bamunu Mudiyanselage, Chunyan Ji, Jie Hu, and Yi Pan. Fast deep learning training through intelligently freezing layers. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1225–1232, 2019.

[17] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 10 2021.

[18] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, volume 27, 2014.