Bulletin of Faculty of Science, Zagazig University (BFSZU)
e-ISSN: 1110-1555
Volume-2025, Issue-1, pp-114-123
https://bfszu.journals.ekb.eg/journal
Research Paper DOI: **10.21608/bfszu.2024.297682.1400**

==========

FT-Transformer for Intrusion Detection in IoT Environment

Ibrahim Ahmed Fares[1] and Mohamed Abd Elaziz[1]
Mathematics Department, Faculty of Science, Zagazig University, Zagazig, 44519, Egypt.
Corresponding author: E-mail: ifares.cs@gmail.com

ABSTRACT : **This work proposes the use of the advanced neural architecture of the Feature Tokenizer (FT)-Transformer for the Intrusion Detection System (IDS) in an IoT environment. By benefiting from the powerful self-attention in transformers, the FT-Transformer captures and identify complex and complicated dependencies and interactions among features in IoT data. We conducted a series of of experiments to evaluate the proposed TF-Transformer for assessing and enhancing. The RT_IOT2022 dataset used in training and evaluating the proposed model. The performance of the model is assessed based on the resulting metrics of accuracy, precision, recall, and F1-score. The experimental results showed that the FT-Transformer improved the performance of cyberattack detection in an IoT network and, in comparison to Deep Learning (DL) models such as CNN, RNN, and autoencoder, could offer high accuracy and robustness in output prediction. Results were found which indicated that FT-Transformer model could have a potential application to improve IoT security and provide robust frameworks for further research and development.**

KEYWORDS: Transformers; IoT; Intrusion Detection; Cybersecurity; AI.

Date of Submission: 14-6-2024 Date of acceptance: 16-07-2024

I. Introduction

The increasing proliferation of Internet of Things (IoT) has indeed transformed the landscape of many sectors through increased connectivity and automation. Nevertheless, with such widespread adoption, IoT systems have been exposed to huge security challenges, positioning them as prime targets for cyber attacks (Saadouni et al., 2024). Intrusion Detection Systems (IDS) are crucial for protecting such networks, since they identify and suppress possible threats in real time. While being effective, classical IDS solutions are often facing multiple challenges in complex and dynamic IoT environments (Liu & Wu, 2023).

Current IDS have some limitations, particularly in the context of the highly dynamic and increasingly complexity of the IoT networked environment (Mohammadi et al., 2018). Classical IDS approaches often struggle with the unique characteristics and challenges presented by IoT environments. One major limitation is the inability to efficiently handle the vast and heterogeneous nature of IoT data. IoT networks generate massive data coming from different devices, often using different protocols and standards, which is overwhelming to analyze and process for the conventional IDS. One huge limitation is that conventional methods of IDS don't have any adaptability. Most current systems depend on predefined rules and signatures, which poses a problem for new and novel threats. With the rapid dynamism of IoT ecosystems in terms of frequent updates and new introductions of devices, it calls for IDS that will learn and keep pace with these new attack patterns. The current systems are weak in this respect (Shiammala et al., 2023).
Besides, the majority of classical IDSs do not make the best use of the advancements in Machine Learning (ML) and AI. While some of these systems include the most basic anomaly detection techniques, they are not equipped with the advanced algorithms needed to catch subtle and complex attack vectors. This becomes a problem, especially in IoT networks, where attackers are likely to use sophisticated strategies to compromise a system (Yin et al., 2017).
Recent advances in ML have shown promise in addressing these challenges. One such contribution is a recent advancement that presents FT-Transformers (Saraniya et al., 2024), a novel model architecture that leverages the power of transformers, originally designed for natural language processing, for enhancing feature representation and

improving detection accuracy. The fact that FT-Transformers can handle heterogeneous data and capture intricate patterns makes it particularly suitable for IoT environments, where data is sourced from various sources and varies in structure and scale.

This paper investigates the application of FT-Transformers for intrusion detection in IoT networks. It seeks to achieve a robust intrusion detection system capable of detecting and responding to cyber threats effectively while taking full advantage of the benefits of transformers. The proposed model is evaluated on the RT_IoT2022 dataset (Airlangga, 2024). The evaluation process compared to state-of-the-art DL models such as CNN (Alzubaidi et al., 2021), RNN (Yin et al., 2017), and Autoencoder (Kunang et al., 2018).

The rest of the paper is constructed as follows; Section II provide some of related works in IDS area. Section III presents a description of the FT-Transformer model. In Section IV, we presented the proposed methodology of the current work. The experiments and results presented in Section V with deep discussion, Finally, Section VI summarizes the current work and present some of future work.

II. Related Work

Initial research on IDS for IoT networks highlighted the basic issues and the importance of strong security paradigms, (Magott et al., 2007) presented the concept of fault trees with time dependencies for the description of intrusions and timing requirements for IDS. In fact, this approach clearly stated that timely detection of the security breaches can prevent large-scale damage.

The advent of the IoT presented new challenges because of the resource-constrained aspects of the IoT devices. Fu et al. introduced an automata-based IDS approach that was designed to accommodate the unique requirements of the heterogeneous IoT networks. Their model was designed to detect various types of attacks—jamming, false, replay, and so forth—and it was effective in experiments (Fu et al., 2017).

ML techniques were then increasingly used in the development of more powerful IDS. For example, (Anthi et al., 2019) created a supervised IDS focusing on IoT equipment in smart homes. This could classify behavior performed by IoT devices, while offering great accuracy in specifying the appearance of malicious network activity and attack types (Anthi et al., 2019). This potential was further investigated by (Yang et al., 2018) through active learning for the intrusion of wireless IoT in detection, showing great potential in combining human intelligence with ML to improve detection accuracy (Yang et al., 2018).

The most recent developments involve the use of transformer-based models in IDS. An improvement of the transformer model to overcome the issues of long training time and low accuracy of detection in data classes was done by (Liu & Wu, 2023). Their approach was integrated with several novelties in data processing strategies and position encoding methods to enhance performance. Similarly, (Wang et al., 2023) also developed TransIDS, a transformer-based approach to deliver a multi-headed self-attention mechanism that extends the generalization and feature extraction abilities of IDS in IoT.

The latest was in the development of the transformer model for real-time and robust IDS applications. To this end, (Ahmed et al., 2023) introduced an MTNN model in response to the needs of reliable improvement in the detection performance of existing approaches. The method combined traditional LSTM and RNN techniques with advanced transformer models to increase accuracy and robustness in IoT network intrusion detection. **Table 1** presents the summary of the related work.

Table 1: Related Work Summary

Authors	Model Used	Advantages	Disadvantages
(Magott et al., 2007)	(FTTD)	Timely detection of security breaches	Requires detailed timing analysis
(Fu et al., 2017)	Automata-based IDS	Effective detection of various attack types	Complex implementation for heterogeneous IoT networks
(Anthi et al., 2019)	Supervised IDS	High accuracy in classifying IoT device behavior and detecting attacks	Limited to smart home environments
(Yang et al., 2018)	Active Learning	Improved detection accuracy by integrating human intelligence with ML	Still in its infancy, requiring more research
(Liu & Wu, 2023)	Improved Transformer Model	Enhanced performance with novel data processing and position encoding methods	Lengthy training times and low detection accuracy for overlapping classes
(Wang et al., 2023)	TransIDS	Improved generalization and feature extraction capabilities	Dependent on balanced datasets for optimal performance
(Abosata et al., 2023)	FT-CID	High detection accuracy through combination of local and global parameters	Complexity in federated learning implementation
(Ahmed et al., 2023)	MTNN	Significant improvements in detection performance across various metrics	Requires high computational resources
(Ding et al., n.d.)	TMG-GAN	Effective on imbalanced data; High precision and recall	Complex training process; Likely issues with data generation
(Vaiyapuri et al., 2024)	STL with SSAE & LSTM	High accuracy (86.31%); Efficient dimensional reduction	Needs large amounts of labeled data; Complicated architecture

III. Background on Feature Tokenizer (FT)-Transformer

The Feature Tokenizer (FT)-Transformer is the advanced neural network architecture, specially designed for dealing with structured data, like the tabular data residing in most IoT applications. It makes use of the strong self-attention mechanism in transformers for the learning of complex relationships and interactions among the features. The FT-Transformer is particularly effective for scenarios where traditional ML models perform poorly because of high dimensionality or the requirement to capture long-range dependencies among the features.

Key Components and Mathematical Formulation

1. **Input Embedding:**
   The first step in the FT-Transformer involves transforming the input features into a suitable representation. Each feature x_i in the input vector $\mathbf{x} \in R^d$ is embedded into a higherdimensional space using a learnable embedding matrix $\mathbf{E}$.
   $$\mathbf{e}_i = \mathbf{E} \cdot x_i$$
   where $\mathbf{e}_i$ is the embedded representation of the i-th feature.
2. **Feature Tokenization:**
   The embedded features are then tokenized, creating a sequence of feature tokens $\mathbf{T}$. This tokenization allows the transformer to treat each feature as a distinct entity, enabling the model to capture interactions between different features.
   $$\mathbf{T} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_d]$$
3. **Self-Attention Mechanism:**
   The core of the FT-Transformer is the self-attention mechanism, which allows the model to weigh the importance of each feature token relative to others. The attention mechanism computes a weighted sum of the input tokens, where the weights are determined by the similarity between tokens.
   $$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$
   where $\mathbf{Q}$ (queries), $\mathbf{K}$ (keys), and $\mathbf{V}$ (values) are linear transformations of the input tokens $\mathbf{T}$:
   $$\mathbf{Q} = \mathbf{T}\mathbf{W}_Q,\ \mathbf{K} = \mathbf{T}\mathbf{W}_K,\ \mathbf{V} = \mathbf{T}\mathbf{W}_V$$
   Here, $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable weight matrices.
4. **Multi-Head Attention:**

To enhance the model's ability to focus on different parts of the input, the FT-Transformer employs multi-head attention, which runs several self-attention mechanisms in parallel and concatenates their outputs.

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{ head }_1, \text{ head }_2, \ldots, \text{ head }_h]\mathbf{W}_O$$

where each attention head head $_i$ is computed as:

$$\text{head }_i = \text{ Attention }\left(\mathbf{Q}\mathbf{W}_Q^i, \mathbf{K}\mathbf{W}_K^i, \mathbf{V}\mathbf{W}_V^i\right)$$

and $\mathbf{W}_O$ is a learnable output projection matrix.

5. **Feed-Forward Network:**
After the attention mechanism, the tokens are passed through a position-wise feed-forward network, which consists of two linear transformations with a ReLU activation in between.

$$\text{FFN}(\mathbf{T}) = \text{ReLU}(\mathbf{T}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are learnable parameters.

6. **Output Layer:**
Finally, the output of the feed-forward network is passed through a linear layer to produce the final prediction.

$$\hat{y} = \mathbf{W}_{\text{out}}\,\mathbf{T} + \mathbf{b}_{\text{out}}$$

where $\mathbf{W}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ are the weights and bias of the output layer, respectively.

Advantages and Applications

An FT-Transformer is particularly suited for IoT applications with heterogeneous and high-dimensional data, where dependencies and interactions are complex. Therefore, it is particularly good in handling very large datasets, capturing complex dependencies and interactions among features in structural data with very high accuracy and robustness, giving state-of-the-art performance improvement on intrusion detection systems.

IV. Proposed Methodology

In this section, we outline the proposed methodology for utilizing FT-Transformers toward enhancing intrusion detection in IoT networks. In a nutshell, the methodology follows a number of key stages: data preprocessing, model architecture design, training and validation, and evaluation. Every stage will be critical for the robustness and effectiveness of the proposed IDS.

Data Preprocessing

The first step involves preprocessing the IoT network traffic data to prepare it for input into the FT-Transformer model. This process includes:

1. Data Collection: Gathering data from various IoT devices and sensors, ensuring it represents a wide range of normal and malicious activities.

2. Normalization: Standardizing the data to ensure that all features contribute equally to the model's learning process. This involves scaling numerical features to a uniform range.

3. Feature Engineering: Extracting relevant features that capture the essential characteristics of network traffic. This may involve creating new features through aggregation or transformation of existing ones.

4. Dimensionality Reduction: Applying techniques such as Principal Component Analysis (PCA) to reduce the feature space, thereby minimizing computational complexity while retaining significant information.

Model Architecture Design

The core of the proposed methodology is the FT-Transformer model, designed to effectively handle the complexities of IoT network data. The architecture includes:

1. Input Layer: Receiving the preprocessed data as input.

2. Embedding Layer: Converting the input features into dense vectors that can be processed by the transformer layers.

3. Transformer Layers: Utilizing multiple transformer layers to capture complex patterns and dependencies in the data. Each transformer layer consists of multi-head self-attention mechanisms and feed-forward neural networks.

4. Output Layer: Producing a classification output that indicates whether the input represents normal or malicious activity. This is typically implemented using a *softmax* layer for binary classification.

Training and Validation

The training process involves optimizing the FT-Transformer model to accurately classify network traffic. Key steps include:
1. Data Splitting: Dividing the dataset into training, validation, and test sets to ensure unbiased evaluation of the model.
2. Loss Function: Using a suitable loss function, such as binary cross-entropy, to guide the optimization process.
3. Optimization: Employing gradient descent-based optimization algorithms, such as Adam, to minimize the loss function and improve model performance.
4. Regularization: Applying techniques such as dropout and weight decay to prevent overfitting and enhance generalization.

Evaluation
The final stage involves evaluating the performance of the trained FT-Transformer model. This includes:
1. Metrics: Utilizing standard metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve to assess model performance.
2. Benchmarking: Comparing the FT-Transformer model with traditional IDS methods and other ML-based approaches to demonstrate its superiority.
3. Robustness Testing: Ensuring the model's robustness by testing it against various types of attacks and scenarios commonly encountered in IoT environments.

By following this methodology, we aim to develop a highly effective IDS that leverages the strengths of FT-Transformers to enhance the detection and mitigation of cyber threats in IoT networks. This approach not only addresses the limitations of current IDS methods but also sets a new benchmark for future research in IoT security.

V. Experiments and Discussion

In this study, the RT_IoT2022 dataset was classified by the FT-Transformer model for the binary classification of IoT. Specifically created to replicate real-world IoT scenarios, encompassing both benign and malevolent activity. The dataset was divided into test, validation, and training sets to allow for a thorough assessment of the model's functionality. The dataset is explained in the subsection as follows.

Description of the RT_IoT2022 Dataset
The UCI ML Repository offers the RT-IoT2022 dataset, a comprehensive dataset collected from real-time IoT infrastructure. It captures both friendly and hostile activities by integrating data from a range of IoT devices and including complex network assault scenarios. This dataset includes devices like MQTT-Temp, Wipro-Bulb, and ThingSpeak-LED to simulate real-world IoT scenarios. It contains many Nmap scan kinds, DDoS (using Hping and Slowloris), and Brute-Force SSH simulated assaults. The Zeek network monitoring tool and the Flowmeter plugin are used to carefully record the network traffic statistics, giving a comprehensive picture of the intricate nature of IoT network traffic.

Table ***2*** summarized the main characteristics of the dataset.

Table 2: Main Characteristics of RT-IoT2022

Aspect	**Description**
Source	Real-time IoT infrastructure
Devices Included	ThingSpeak-LED, Wipro-Bulb, MQTT-Temp
Attack Scenarios	Brute-Force SSH, DDoS (Hping, Slowloris), Nmap (various scan types)
Network Monitoring Tools	Zeek with Flowmeter plugin
Number of Instances	123,117
Number of Features	83
Data Types	Tabular, Sequential, Multivariate
Tasks	Classification, Regression, Clustering
Feature Types	Real, Categorical
Missing Values	No

Results And Discussion

With the highest accuracy (98.50%) and similarly remarkable precision and recall rates (98.46% and 98.50%, respectively), the FT-Transformer is at the top of the group. With an F1 score of 98.43%, this model exhibits a high degree of accuracy in classifying or predicting events, while also minimizing the trade-off between precision and recall. To obtain better performance, this model probably makes use of transformer architectural benefits such attention mechanisms.

Table 3 shows a study of the FT-Transformer's performance in comparison to more conventional DL models such as RNNs, CNNs, and Autoencoders for an unidentified job that is probably linked to classification or prediction. Based on four important metrics—Accuracy, Precision, Recall, and F1 score—the table assesses these models.

- **RNN:** Performs admirably, achieving an F1 score of 95.93% with accuracy of 96.5%, precision of 94.55%, and remarkably high recall of 97.34%. Because of their high recall rate and capacity to handle sequential input, RNNs are often preferred.
- **CNN:** This model records values of 97.01% and 97.09%, respectively, slightly outperforming the RNN in terms of accuracy and precision. With a similar F1 score of 97.08%, the recall rate is quite similar to the RNN's 97.06%. CNNs are renowned for their ability to interpret spatial data, which is useful for problems involving image or signal processing.
- **Autoencoders:** They achieve an accuracy of 97.39% and a balanced precision and recall of 97.43% and 97.39%, respectively, and are mostly employed for data compression and reconstruction. With an F1 score of 97.39%, the model's performance is well-balanced and reflects both precision and recall.

- **FT:** With the highest accuracy (98.50%) and similarly remarkable precision and recall rates (98.46% and 98.50%, respectively), the FT-Transformer is at the top of the group. With an F1 score of 98.43%, this model exhibits a high degree of accuracy in classifying or predicting events, while also minimizing the trade-off between precision and recall. To obtain better performance, this model probably makes use of transformer architectural benefits such attention mechanisms.

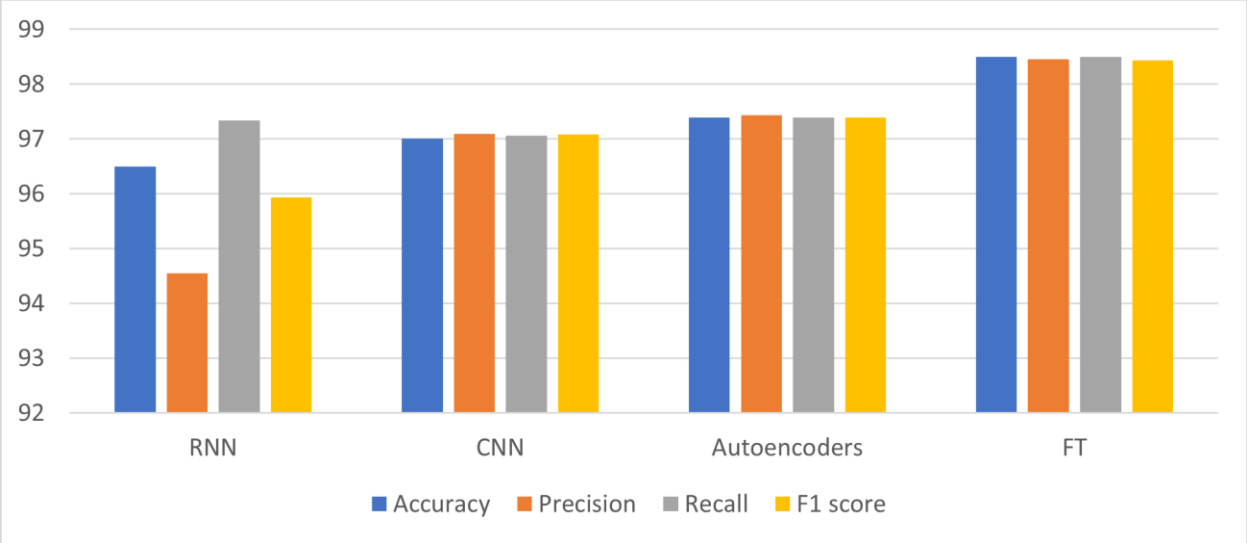
Table 3: Performance Results of the FT-Transformer compared to DL models

Model	Accuracy	Precision	Recall	F1 score
RNN	96.5	94.55	97.34	95.93
CNN	97.01	97.09	97.06	97.08
Autoencoders	97.39	97.43	97.39	97.39
FT	**98.50**	98.46	**98.50**	**98.43**

Overall, the FT-Transformer showcases the best performance across all metrics, indicating its robustness and efficiency in handling the tasks compared to other DL models.

The comparative performance metrics of four alternative models—RNN, CNN, Autoencoders, and FT-Transformer—across four important dimensions—Accuracy, Precision, Recall, and F1 Score—are visually represented by the bar chart (Figure 1). The FT-Transformer model has the best results across the board, indicating its exceptional handling of tasks that are probably connected to categorization. It is noteworthy for achieving the best F1 Score (98.43%), Accuracy (98.5%), Precision (98.46%), and Recall (98.5%) when compared to the other models. Even if the other models are working well, their overall metrics are a little bit lower; RNNs are somewhat behind CNN and Autoencoders in performance. This graph does a good job of comparing and highlighting the benefits of the FT-Transformer, highlighting its potential for use in applications that need a high degree of precision and dependability.

Figure 1: Performance metric comparison between the FT-Transformer and DL models



The accuracy curves for four models—RNN, CNN, Autoencoders, and FT-Transformer—over 20 epochs are shown in **Figure 2**, which illustrates how each model improves in terms of learning accuracy. Throughout the epochs, the FT-Transformer—shown in red—performs better than the others, keeping a constant lead while improving

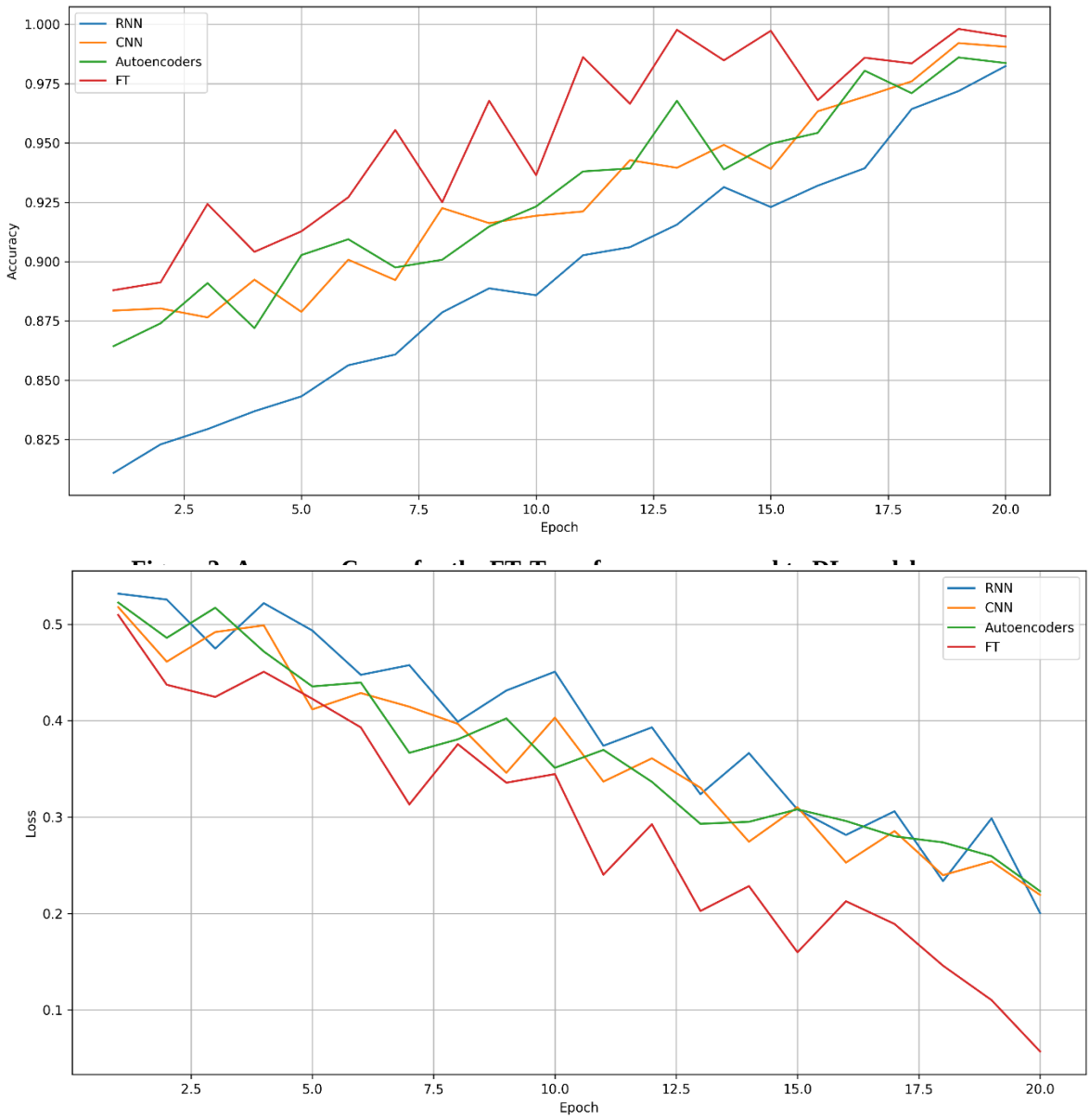


Figure 3: Loss Curve for the FT-Transformer compared to DL models

precision. This shows that it can learn and generalize well, as evidenced by the fact that it eventually achieved the greatest accuracy of all the models evaluated. On the other hand, the green line, which represents the Autoencoders, shows some fluctuation but typically trends higher, indicating learning that is successful but not very constant. Both the RNN (blue line) and CNN (orange line) exhibit improvements with time, but the RNN's path is characterized by more noticeable variations since it begins with a lower accuracy than the other two.

This **Figure 3** shows the loss curves for the 20 epochs of four models: RNN, CNN, Autoencoders, and FT-Transformer. It shows how each model optimizes the learning process over time. The FT-Transformer, in red, is better than all the other models in the sense that it has a sharp decrease in loss, keeping that line consistent up to the end of the epochs, with the lowest values obtained. This trend indicates its effectiveness in quickly reducing errors to indicate high efficiency in the optimization process. The other models—Autoencoders in green, CNN in orange,

RNN in blue—show a drop in their loss curves but less steep with a lot of zigzags than the FT-Transformer. The RNN model shows this drop as the slowest one, making it less efficient in optimization over epochs. The CNN and Autoencoders follow quite a similar pattern of drop, while the Autoencoders have some ups and downs, which might indicate some setbacks during the training of the model.
Overall, the FT-Transformer minimizes loss fast and effectively, with the corresponding superiority of robustness and higher accuracy in tasks.

In conclusion, the analysis of FT-Transformer against DL models like RNNs, CNNs, and Autoencoders has shown the superiority of the FT-Transformer in the accuracy and loss metrics. Overall, the FT-Transformer outperformed all other models in terms of both highest achievable accuracy and the lowest loss over several epochs, thereby proving that it can handle more complex learning tasks. These results not only showcase that the FT-Transformer excels at optimization and generalization from data but also allows for applying the proposed approach in a larger set of challenging real-world problems. The FT-Transformer can be a potential deployment in extremely reliable and accurate-based systems, specifically in areas where minute and robust model performance is required.

VI. Conclusion And Future Work

In all these experiments, the FT-Transformer proves to be vastly successful in improving intrusion detection in IoT networks compared with traditional models, such as RNNs, CNNs, and Autoencoders, in all accuracy and loss metrics. The ease with which it processes large-scale data makes it particularly well-suited to real-time environments where quick and accurate threat detection is of utmost importance. The high accuracy and minimal loss of the FT-Transformer underline its potential to effectively beef up security within IoT systems.
In this direction, some promising ways of extension of the work done on the FT-Transformer are in terms of multi-class classification for more network threat insights, the combination of the model with technologies like federated learning to boost privacy in decentralized systems, and the real-time adaptability of the model to new threats. Also, using the FT-Transformer in other critical domains, such as healthcare or finance, and optimizing it for deployment on resource-scarce devices, will further increase its utility and impact. Such steps could end up making the FT-Transformer more of a general-purpose tool, ready to face the new challenges of cybersecurity in diverse environments.

References

Abosata, N., Al-Rubaye, S., & Inalhan, G. (2023). Customised Intrusion Detection for an Industrial IoT Heterogeneous Network Based on Machine Learning Algorithms Called FTL-CID. *Sensors*, *23*(1). https://doi.org/10.3390/s23010321

Ahmed, S. W., Kientz, F., & Kashef, R. (2023). A modified transformer neural network (MTNN) for robust intrusion detection in IoT networks. *2023 International Telecommunications Conference (ITC-Egypt)*, 663–668.

Airlangga, G. (2024). Comparative Analysis of Machine Learning Models for Intrusion Detection in Internet of Things Networks Using the RT-IoT2022 Dataset. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 656–662. https://doi.org/10.57152/malcom.v4i2.1304

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamar\'\ia, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, 1–74.

Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., & Burnap, P. (2019). A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal*, *6*(5), 9042–9053.

Ding, H., Sun, Y., Huang, N., … Z. S.-I. T. on, & 2023, undefined. (n.d.). TMG-GAN: Generative Adversarial Networks-Based Imbalanced Learning for Network Intrusion Detection. *Ieeexplore.Ieee.OrgH Ding, Y Sun, N Huang, Z Shen, X CuiIEEE Transactions on Information Forensics and Security, 2023•ieeexplore.Ieee.Org*. Retrieved June 5, 2024, from https://ieeexplore.ieee.org/abstract/document/10312801/

Fu, Y., Yan, Z., Cao, J., Koné, O., Cao, X., & others. (2017). An automata based intrusion detection method for internet of things. *Mobile Information Systems*, *2017*.

Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., & others. (2018). Automatic features extraction using autoencoder in intrusion detection system. *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, 219–224.

Liu, Y., & Wu, L. (2023). Intrusion Detection Model Based on Improved Transformer. *Applied Sciences*, *13*(10), 6251.

Magott, J., Skrobanek, P., & Woda, M. (2007). Analysis of timing requirements for intrusion detection system. *2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX'07)*, 278–285.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923–2960.

Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., & Khacha, A. (2024). Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature. *Cluster Computing*, 1–27. https://doi.org/10.1007/S10586-024-04388-5/METRICS

Saraniya, S., Sowmiya, M. V., Kalpana, B. N., & Karthi, M. (2024). Securing Networks: Unleashing the Power of the FT-Transformer for Intrusion Detection. *ICCECE 2024 - International Conference on Computer, Electrical and Communication Engineering*. https://doi.org/10.1109/ICCECE58645.2024.10497252

Shiammala, P. N., Duraimutharasan, N. K. B., Vaseeharan, B., Alothaim, A. S., Al-Malki, E. S., Snekaa, B., Safi, S. Z., Singh, S. K., Velmurugan, D., & Selvaraj, C. (2023). Exploring the artificial intelligence and machine learning models in the context of drug design difficulties and future potential for the pharmaceutical sectors. *Methods*.

Vaiyapuri, T., Binbusayyis, A., Kharj, A., & Arabia, S. (2024). Deep self-taught learning framework for intrusion detection in cloud computing environment. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *13*(1), 747–755. https://doi.org/10.11591/IJAI.V13.I1.PP747-755

Wang, P., Wang, X., Song, Y., Huang, J., Ding, P., & Yang, Z. (2023). TransIDS: A Transformer-based approach for intrusion detection in Internet of Things using Label Smoothing. *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, 216–222.

Yang, K., Ren, J., Zhu, Y., & Zhang, W. (2018). Active learning for wireless IoT intrusion detection. *IEEE Wireless Communications*, *25*(6), 19–25.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, 21954–21961.