Transformer Model for Intelligent Network Intrusion Detection System

Md Mahmudul Alam Imon
Department of Management Information Systems
University of Dhaka
Dhaka, Bangladesh
mahmudulalam.imon@gmail.com

***Abstract*—This paper presents a TabTransformer-based Network Intrusion Detection System (NIDS) to handle the problems of identifying hostile network traffic patterns in the CIC-IDS2017 dataset. Borderline-SMote and RandomUnderSampler techniques were applied to balance the dataset, which showed a clear class imbalance (84.92% benign and 15.08% attack samples). The Cross-Entropy Loss function was thus given a class weight of [1.0, 5.0] that gave minority class samples priority during training. Using Multi-Head Self-Attention (MHSA) and Feed-Forward Networks (FFN), the TabTransformer architecture effectively recorded complicated relationships between features in both numerical and categorical data. Using an Adam optimizer with a learning rate of 0.0001, the model trained for ten epochs with great consistency. With an accuracy of 98.56%, precision of 97.38%, recall of 99.80%, F1 score of 98.58%, and an AUC-ROC score of 0.9987, the performance analysis produced quite remarkable outcomes. Confusion matrix heatmaps, classification report heatmaps, and training-validation curves among other visualizations revealed constant learning patterns with little incorrect classifications. These results show the suitability of the TabTransformer for network intrusion detection operations, therefore providing a dependable and balanced solution for cybersecurity systems. .**

***Index Terms*—Network Intrusion Detection System, Deep Learning, Transformer, TabTransformer, SMOTE, Multi-Head Self-Attention (MHSA), Feed-Forward Networks (FFN)**

I. INTRODUCTION

Over 60% of the global population uses the internet [1]. The Internet offers several benefits to consumers; however, it also presents substantial risks of cyber-attacks that result in billions of dollars in losses. NIDS (Network Intrusion Detection Systems) may prevent illegal attempts by categorizing malicious access inside network traffic. NIDS systems utilize several learning models, including SVM and Random Tree, to categorize attacks. Nevertheless, certain models exhibit inadequate performance when presented with extensive and complicated datasets and their complex features [2]. Deep Learning has demonstrated its efficacy in effectively handling extensive datasets in natural language processing (NLP) and image processing.

There have been several researches and experiments on NIDS using different deep learning models such as CNN and transformer-based models. Demóstenes et al. experimented with TabNet, a model that uses an attention mechanism similar to that of a transformer, and achieved an accuracy of 97% on CIC-IDS 2017, 95% on CSE-CICIDS 2018, and 98% on CIC-IDS 2019 datasets [3]. Leila et al experimented with CNN with an accuracy rate of 98.08% on CIC-IDS 2017 dataset [4].

In recent years, self-attention models, such as transformers and their derivatives, have demonstrated significant success in classifying texts, machine translation, and many NLP applications. Following these achievements, researchers are now utilizing transformer-based models for intrusion detection, attaining satisfactory outcomes [5] [6] [7]. Some of the popular transformer models include generative pre-trained transformer (GPT), bidirectional encoder representations for transformers (BERT), T5, etc.

This project proposes a transformer-based approach for network intrusion detection, utilizing the TabTransformer architecture, which is optimized for processing tabular data by utilizing Multi-Head Self Attention and Feed-Forward Networks to capture both numerical and categorical feature relationships. The model is trained and validated on the well-known CIC-IDS 2017 dataset, a benchmark dataset for assessing Network Intrusion Detection Systems (NIDS) addressing class imbalance using Borderline-SMOTE and class-weighted Cross-Entropy Loss. The assignment was structured as a binary classification task that involved differentiating between attack and benign traffic. Experimental results indicate that the model attained remarkable performance, especially in detecting attack instances, with a recall of 99.80% and an overall accuracy of 98.56%. Moreover, the model achieved an F1 score of 98.58%, a precision of 97.38%, and a Matthews Correlation Coefficient (MCC) of 97.15%, highlighting its robustness and efficacy in intrusion detection tasks.

II. RELATED WORKS

Advances in Transformer-Based NIDS: New studies have shown how transformer designs might improve NIDS performance. Using TabTransformer, a transformer-based architecture designed for network intrusion detection, Wang et al. (2024) suggested a binary classification framework. Their research underlined how well the model could manage tabular data, hence improving the accuracy on benchmark datasets [8].

Likewise, a new multi-scale network intrusion detection system called IDS-MTran was presented to collect multi-scale traffic features, so extending the detection capacity of NIDS by means of transformers. This method emphasizes

how adaptable transformer models are in changing to different network traffic patterns [9].

A transformer-based NIDS algorithm was developed in the framework of cloud environments to solve the particular difficulties presented by cloud infrastructures. This model uses transformers' attention mechanism to identify anomalies in cloud network traffic, thus improving detection rates and lowering false positives [10].

Moreover, it has been investigated how to address class imbalance in NIDS by combining transformer models with other deep learning architectures. Combining the capabilities of both architectures to increase detection accuracy—especially for minority classes, an advanced hybrid Transformer-CNN deep learning model was suggested [11].

Comparative Research Using Conventional Models: Although conventional machine learning models such as Support Vector Machines (SVM) and Decision Trees have been used in NIDS, they sometimes find it difficult to capture complicated feature interactions and manage big-scale, imbalanced datasets. By means of their attention mechanisms, transformer-based models, on the other hand, may replicate complex data linkages, so enhancing the detection performance. For instance, the FlowTransformer framework detects complicated incursion patterns by using transformer designs to identify long-term network behaviors, hence surpassing conventional approaches.

Correcting Class Inbalance and Real-Time Detection: In NIDS, class imbalance is still a major obstacle where some attack strategies are underrepresented. Synthetic Minority Oversampling Technique (SMote) and transformer-based models have been merged to help with this problem. Furthermore investigated is the creation of real-time network intrusion detection systems with decision transformers in order to strike a compromise between detection accuracy and quick responses [12].

Including transformer designs in NIDS shows a good path to improve detection capacity. These models provide flexibility to changing network settings and solve constraints inherent in conventional techniques including handling difficult feature interactions and class imbalances. Future studies should keep looking at how transformers might be used with other deep learning models and real-time detection systems to improve NIDS performance even more [13].

III. Methodology

A. Dataset and Features

In this project, the model is trained with the CIC-IDS 2017 dataset, a widely used dataset to train the NIDS models. The dataset has over 2 million samples with 77 features. Some of the important features are Protocol, Packet Length, Total Forward Packets, Avg Packet Size, etc. The predicted label has 15 classes, benign and 14 attack types including DoS Hulk, DDoS, DoS GoldenEye, FTP-Patator, etc.

Fig. 1: Binary Distribution-Class Imbalance

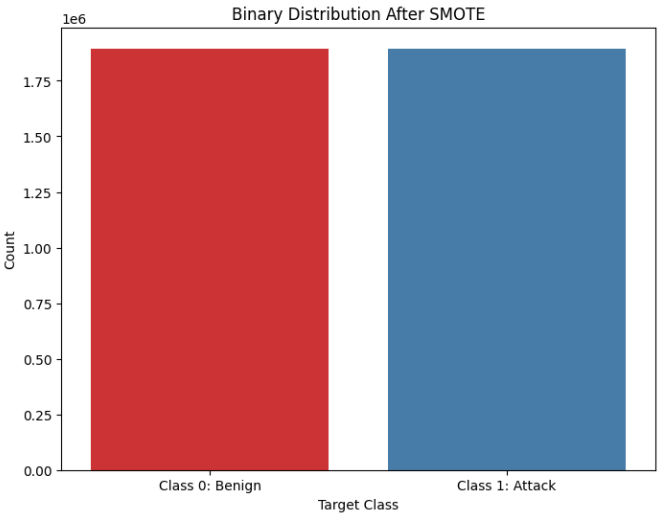

Fig. 2: Balanced Class after SMOTE

B. Data Preparation

The dataset is highly imbalanced with 84.92% benign and 15.08% attack classes (Figure 1). To address this imbalance, a Binary Classification approach was adopted to classify traffic into Benign and Attack categories. To further mitigate the higher class imbalance, BorderlineSMOTE, and RandomUnderSampler were used. BorderlineSMOTE generates synthetic samples for minority classes and RandomUnderSampler removes some unnecessary extra samples from the majority class. This combined approach resulted in a more balanced dataset, enhancing the model's ability to generalize effectively across both classes (Figure 2).

C. Resolving Class Imbalance Using SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is used to fix the class imbalance in the dataset before training the TabTransformer model. The dataset indicates a significant

imbalance, including 84.92% benign samples and only 15.08% attack samples. SMOTE generates synthetic samples for the minority class instead of replicating existing ones. Borderline-SMOTE (kind='borderline-1') is applied, focusing on the creation of synthetic samples near the decision boundary, where minority class samples are most susceptible to misclassification.

Mechanism of SMOTE:

Determine Nearest Neighbors: SMOTE identifies the k nearest neighbors from the same class for each sample in the minority class.

Generate Synthetic Samples: A synthetic sample is produced by combining the minority sample and one of its nearest neighbors utilizing the formula:

$$x_{\text{new}} = x_i + \lambda \cdot (x_{\text{neighbor}} - x_i)$$

where:

x_i is the original minority sample.

neighbor $x_neighbor$ is one of the nearest neighbors.

λ is a random value between 0 and 1.

Focus on Borderline Samples: Borderline-SMOTE generates synthetic samples along the decision boundary to enhance the model's ability to differentiate between classes in difficult areas.

Balance the Dataset: Following the generation of synthetic samples by SMOTE, RandomUnderSampler is used to get a balanced dataset by decreasing the number of majority class samples [14] [**?**].

D. Model Architecture

The proposed NIDS system uses TabTransformer, a transformer-based model, that is particularly designed for processing tabular data where both categorical and continuous features are handled efficiently. The TabTransformer architecture begins with Column Embedding, converting categorical characteristics into dense vector representations (Figure 3). The first layer of the model is the input embedding layer, which increases the representational capacity of the raw input features by mapping them into a higher-dimensional hidden space through a linear transformation. The embeddings are then fed into a Transformer Encoder, that includes two layers. Each layer uses multi-head self-attention to capture relationships among features, succeeded by feed-forward networks that refine and diminish the feature dimensions. Normalization techniques, such as layer normalization and dropout, are integrated into the encoder to enhance training stability and prevent over-fitting. The sequence is then reduced to a single feature vector by combining the encoded feature representations using global average pooling. A fully connected linear classifier processes the combined representation and makes predictions for the binary classification task [15]. By giving the minority class more weight, weighted cross-entropy loss is used to reduce the dataset's large class imbalance. The model utilizes the Adam optimizer, with training and validation losses and accuracies evaluated across ten epochs to ensure consistent performance. This design integrates the representational capabilities of Transformers with methods to address the class imbalance and over-fitting in tabular data [16] [17].

Feature Embedding: Numerical and categorical features are mapped into dense vectors using a linear embedding layer:

$$E = W \cdot X + b$$

where *X* is the input feature vector, *W* is the weight matrix, and *b* is the bias vector.

Transformer Encoder Layers: The embedded vectors are passed through multiple Transformer encoder layers. Each encoder layer has two main components: **Multi-Head Self-Attention (MHSA):**

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where *Q,K,V* represent query, key, and value matrices, and *dk* is the dimension of the key.

Feed-Forward Network (FFN):

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Global Pooling and Classification Layer: After passing through the Transformer layers, global average pooling is applied:

$$z = \frac{1}{N}\sum_{i=1}^{N} h_i$$

where *hi* are the Transformer outputs. A final linear layer maps the pooled representation to class probabilities.

E. Model Training

The TabTransformer model is trained by optimizing its performance over ten epochs using the Adam optimizer with a learning rate of 0.0001 and the Cross-Entropy Loss function. A weighted loss function with class weights set to [1.0, 5.0] is used to correct for class imbalance. The model has several Transformer Encoder layers after an embedding layer that converts the input features into dense representations. Each Transformer layer has a Multi-Head Self-Attention (MHSA) mechanism and a Feed-Forward Network (FFN), with Layer Normalization and Dropout (0.1) implemented subsequent to each sub-layer. The MHSA encodes complex feature interactions, whereas the FFN combines non-linearity and feature transformations. The feature representations are combined by global average pooling after processing through these layers, and these pooled features are then mapped to the output classes by a final linear classification layer. A batch size of 128 is used to train the model, and data is run through the network during both the training and validation stages. The model's convergence and generalization capabilities are assessed at each epoch using performance indicators, such as training loss and validation loss (Figure 4).

Loss Function: The model uses Cross-Entropy Loss for classification:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

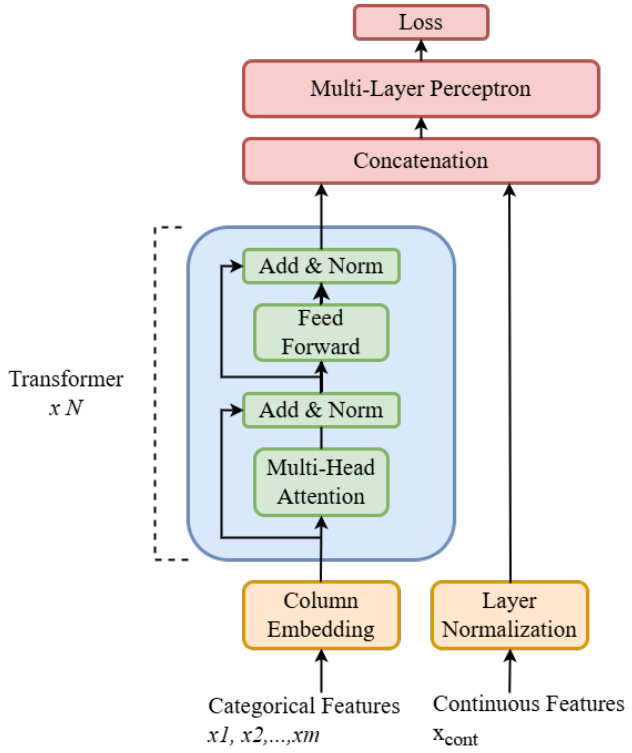

Fig. 3: TabTransformer Architecture

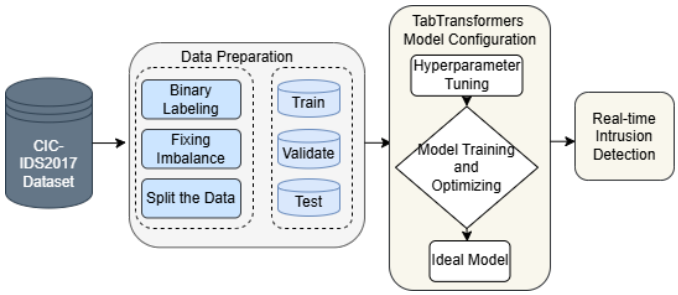

Fig. 4: TabTransformer Training Diagram

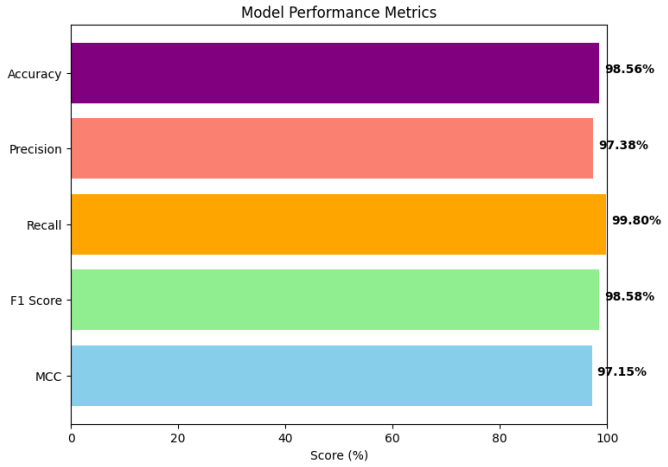

Fig. 5: Model Performance Metrics

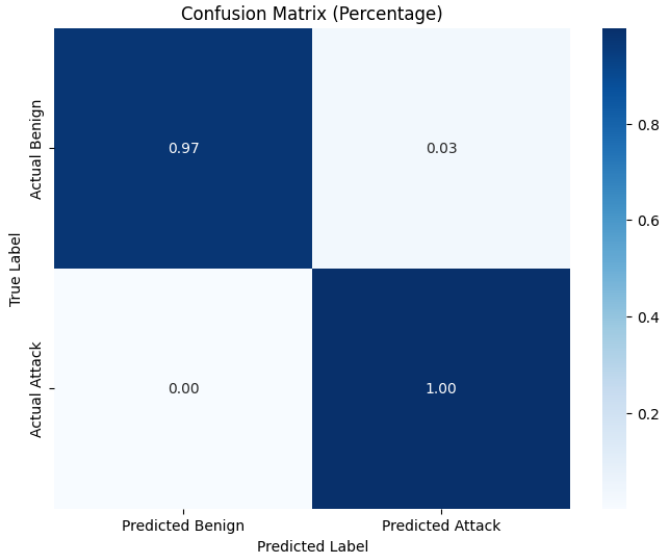

Fig. 6: Confusion Matrix TabTransformer

where y_i is the true label and $\hat{y}_i$ is the predicted probability.

Optimizer: The Adam optimizer updates weights:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{m_t}{\sqrt{v_t} + \epsilon}$$

where:

θ = Model parameters
η = Learning rate
m_t = Moving average of gradients
v_t = Moving average of squared gradients

IV. Results

The TabTransformer model demonstrated excellent performance on the CIC-IDS2017 dataset, achieving consistent improvements across 10 epochs with a learning rate of 0.0001. The training loss steadily decreased, and the validation loss stabilized, indicating effective convergence without signs of overfitting (Figure 9). The final training accuracy reached 98.63%, while validation accuracy peaked at 98.73%, showcasing the model's robustness (Figure 8). Evaluation of the test dataset revealed strong performance, with an accuracy of 98.56%, precision of 97.38%, recall of 99.80%, F1 score of 98.58%, MCC of 0.971, and an AUC-ROC score of 0.9987 (Figure 5). Visualizations such as training and validation loss curves and accuracy plots depicted smooth learning progress, while the confusion matrix heatmap highlighted minimal misclassifications and high predictive accuracy (Figure 6). Furthermore, the classification report heatmap and metric comparison bar plot provided an intuitive overview of the model's balanced performance across all evaluation criteria (Figure 7). These results demonstrate the TabTransformer's ability to effectively handle complex relationships within the dataset, making it highly suitable for network intrusion detection tasks.

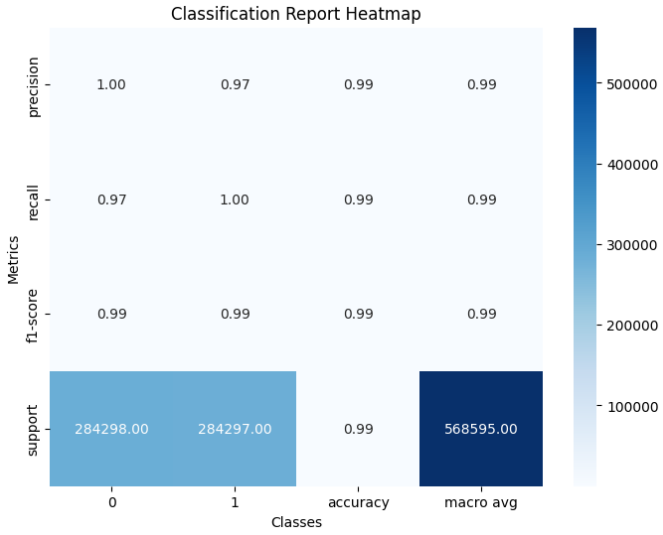

Fig. 7: Classification Report Heatmap

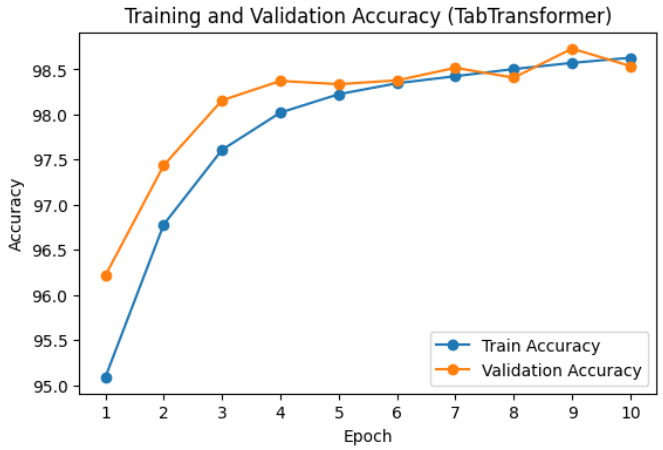

Fig. 8: Training and Validation Accuracy (TabTransformer)

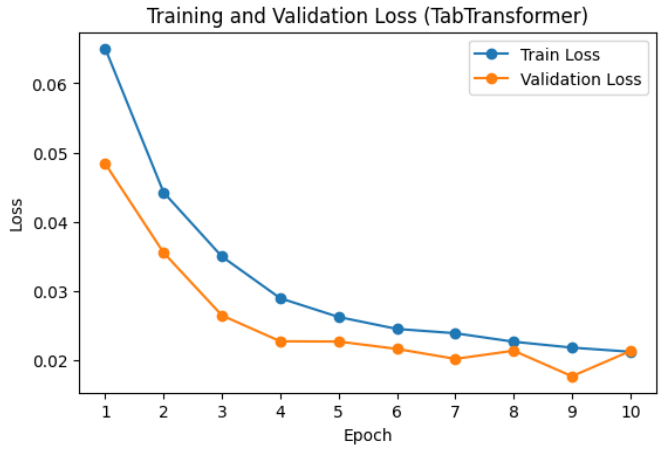

Fig. 9: Training and Validation Loss (TabTransformer)

V. Discussion

The TabTransformer model demonstrated remarkable efficacy in addressing the network intrusion detection issue with the CIC-IDS2017 dataset. A major problem in this dataset was class imbalance, with Benign traffic substantially exceeding Attack traffic (84.92% compared to 15.08%). To resolve this issue, Borderline-SMOTE was utilized to synthetically produce minority class samples next to the decision boundary, then using RandomUnderSampler to properly balance the dataset. Additionally, to give minority class samples priority during training, class weights of [1.0, 5.0] were added to the Cross-Entropy Loss function. The model architecture successfully learned complicated relationships inside tabular data by utilizing Transformer Encoder layers with Feed-Forward Networks (FFN) and Multi-Head Self-Attention (MHSA). Performance measures like accuracy (98.56%), precision (97.38%), recall (99.80%), F1-score (98.58%), MCC (0.971), and an AUC-ROC score (0.9987) demonstrated the model's robustness and reliability. The training was conducted over 10 epochs with a learning rate of 0.0001. The visualizations, including confusion matrix heatmaps, classification report heatmaps, accuracy curves, and loss curves, demonstrated strong proof of constant learning and minimal misclassification. These findings confirm the TabTransformer's capacity to generalize proficiently on unseen data and produce precise predictions, despite inherent difficulties of class imbalance and complex feature relationships. Future endeavors may include exploring deeper Transformer structures or integrating feature engineering methodologies to enhance performance.

VI. Conclusions

In this paper, I proposed a Transformer-based Network Intrusion Detection System (NIDS) using the TabTransformer model that can process tabular data with both numerical and categorical features. I trained the model with the CIC-IDS 2017 dataset which has a very high class imbalance. I successfully solved the class imbalance issue by using Borderline-SMOTE and RandomUnderSampler. In addition, the allocation of class weights ([1.0, 5.0]) in the Cross-Entropy Loss function prioritized the minority class (attack traffic), reducing misclassification rates. The model design, with Transformer Encoder layers with Multi-Head Self-Attention (MHSA) and Feed-Forward Networks (FFN), effectively represented complicated feature relationships in tabular data. Training for over 10 epochs at a learning rate of 0.0001 achieved effective model integration, visualized by continuous reductions in training and validation losses. The evaluation metrics—accuracy (98.56%), precision (97.38%), recall (99.80%), F1 score (98.58%), MCC (0.971), and AUC-ROC score (0.9987)—highlight the robustness and reliability of the suggested methodology. Visualization instruments, including confusion matrix heatmaps, classification report heatmaps, and metric comparison bar plots, provided clear insights into the model's efficiency. The results show that the TabTransformer model is highly capable of intrusion detection tasks, providing significant enhancements compared to conventional machine learning methods. Future

works may include investigating more advanced Transformer designs, optimizing hyperparameters, and applying complicated feature engineering techniques to improve detection precision and scalability in practical network settings.

VII. Acknowledgment

I highly appreciate Dr. Mohammad Saidur Rahman and Dr. Mohammad Ishtiaque Rahman Abir for their tremendous efforts to teach us the complex machine learning and deep learning topics in this intensive bootcamp that resulted in this successful intrusion detection project using an advanced deep learning model within a short period. I am highly grateful for their guidance and support in accomplishing this project and overcoming the challenges associated with the project. I also appreciate the Department of Management Information Systems of the University of Dhaka for offering this amazing learning opportunity.

References

[1] S. Caton and R. Landman, "Internet safety, online radicalisation and young people with learning disabilities," *British Journal of Learning Disabilities*, vol. 50, no. 1, pp. 88–97, 2022.

[2] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "Rtids: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64 375–64 387, 2022.

[3] D. Zegarra Rodríguez, O. Daniel Okey, S. S. Maidin, E. Umoren Udo, and J. H. Kleinschmidt, "Attentive transformer deep learning algorithm for intrusion detection on iot systems using automatic xplainable feature selection," *Plos one*, vol. 18, no. 10, p. e0286652, 2023.

[4] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of cnn-based network intrusion detection," *Applied Sciences*, vol. 12, no. 16, p. 8162, 2022.

[5] T.-N. Nguyen, K.-M. Dang, A.-D. Tran, and K.-H. Le, "Towards an attention-based threat detection system for iot networks," in *International Conference on Future Data and Security Engineering*. Springer, 2022, pp. 301–315.

[6] M. A. Ouni and F. Jemili, "Enhancing intrusion detection systems with transformer models," *Available at SSRN 4841707*, 2024.

[7] Z. Wang, J. Li, S. Yang, X. Luo, D. Li, and S. Mahmoodi, "A lightweight iot intrusion detection model based on improved bert-of-theseus," *Expert Systems with Applications*, vol. 238, p. 122045, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423025472

[8] X. Wang, Y. Qiao, J. Xiong, Z. Zhao, N. Zhang, M. Feng, and C. Jiang, "Advanced network intrusion detection with tabtransformer," *Journal of Theory and Practice of Engineering Science*, vol. 4, no. 03, pp. 191–198, 2024.

[9] C. Xi, H. Wang, and X. Wang, "A novel multi-scale network intrusion detection model with transformer," *Scientific Reports*, vol. 14, no. 1, p. 23239, 2024.

[10] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, "A transformer-based network intrusion detection approach for cloud security," *Journal of Cloud Computing*, vol. 13, no. 1, p. 5, 2024.

[11] H. Kamal and M. Mashaly, "Advanced hybrid transformer-cnn deep learning model for effective intrusion detection systems with class imbalance mitigation using resampling techniques," *Future Internet*, vol. 16, no. 12, p. 481, 2024.

[12] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "Flowtransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Systems with Applications*, vol. 241, p. 122564, 2024.

[13] J. Chen, H. Zhou, Y. Mei, G. Adam, N. D. Bastian, and T. Lan, "Real-time network intrusion detection via decision transformers," *arXiv preprint arXiv:2312.07696*, 2023.

[14] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the smote algorithm and locally linear embedding," in *2006 8th international Conference on Signal Processing*, vol. 3. IEEE, 2006.

[15] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "Tabtransformer: Tabular data modeling using contextual embeddings," 12 2020.

[16] ——, "Tabtransformer: Tabular data modeling using contextual embeddings," *arXiv preprint arXiv:2012.06678*, 2020.

[17] A. KALIDINDI and M. B. ARRAMA, "A tabtransformer based model for detecting botnet-attacks on internet of things using deep learning," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 13, 2023.