Intrusion Detection Using Deep Learning

1st Santoshkumar Tongli
dept. of Computer Science
Colorado State University
Fort Collins, U.S.A.
tskumar@colostate.edu, github-link

2nd Indrajit Ray
dept. of Computer Science
Colorado State University
Fort Collins, U.S.A.

***Abstract*—In this paper, we discuss the learning models employed in Intrusion Detection Systems (IDS), with a specific emphasis on TabNet, which serves as our foundational model. TabNet facilitates classification decisions by integrating both statistical insights and attention-based features. We aim to address the challenges in deep learning through a structured approach that emphasizes logical reasoning and interpretability within the model. This involves a detailed analysis of feature properties, ensuring that our model not only predicts accurately but also provides insights into the reasoning behind its decisions. Through this paper we will understand how this learning model will help in solving real time Intrusion Detection problem.**

***Index Terms*—IDS - Intrusion Detection System, Deep Learning, TabNet**

I. Introduction

In today's digital era, our dependence on the Internet is paramount. It is as essential as water for sustaining a convenient lifestyle and optimizing our daily activities. Over recent decades, exponential growth in internet usage has been driven by advancements in mobile communications, IoT devices, cloud computing, web technologies, and satellite communications. This surge has inevitably led to an increase in network traffic and, consequently, a heightened risk of cyberattacks. The pervasive nature of internet networks fosters opportunities for malicious entities, motivated by potential gains from internet-based attacks. A 2020 security assessment by the National Technology Security Coalition highlighted a significant uptick in network security concerns, evidenced by approximately 620 million accounts compromised in 2019 alone. The trading of compromised data on the dark web and the shift towards remote work during the COVID-19 pandemic, as reported by the 2020 CIRA Cybersecurity Survey, underscore the urgent need for robust cybersecurity measures.

Cyberattacks often involve attackers sending malicious packets after gaining unauthorized access to systems. Their objectives range from stealing, altering, or destroying critical data to launching unauthorized network packets. These attacks exploit vulnerabilities due to user behavior, system configuration errors, or software flaws. As the sophistication of network attacks increases, traditional rule-based Intrusion Detection Systems (IDSs) are often inadequate, prompting a shift towards more dynamic and adaptive security measures. Machine learning (ML) models have been developed to detect intrusion by learning patterns within input features and making decisions based on these learned features. Popular ML models used in IDS include Random Forests, Decision Trees, SVM, and Gradient Boosting. However, these models, despite their intelligence, are susceptible to overfitting, false positives, and require retraining as data evolves.

The another more advanced approach involves the use of deep learning (DL) methods. DL has emerged as a powerful tool in IDS, transcending traditional methods by leveraging advanced models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and various anomaly detection frameworks. These DL approaches demonstrate a superior ability to generalize from large datasets and detect anomalous traffic with increased accuracy and fewer false positives.

This paper explores the integration of deep learning technologies in enhancing IDS, particularly through context-based learning. We propose a framework that identifies optimal features from a given set, employing a transformer model to extract and interpret these features, followed by the use of a 1x1 convolution operation for making classification decisions.

II. Literature Survey

Considerable research has been conducted on intrusion detection using machine learning techniques. Key advancements include identifying and preventing intrusions by recognizing new trends [1].

Significant contributions to intrusion detection systems (IDS) with machine learning include the work by Kiran et al. (2023) [2], who explored various algorithms, particularly emphasizing the Support Vector Machine (SVM) for its robust generalization capabilities. They utilized the "kddcup99" dataset, which comprises 41 features, and highlighted the importance of data preprocessing—specifically feature extraction and variance calculation—to enhance model performance by effectively classifying different network attack types. But the traditional ML techniques such as SVM and Random Forests, as used by Kiran et al. (2023) and Zhu et al. (2021), are effective for certain types of data but often struggle with high-dimensional, heterogeneous data typical in modern network environments. Some of the other work includes, Shah et al. (2020) [3] work focusing on multiclass classification baselines for anomaly-based network intrusion detection systems,

Identify applicable funding agency here. If none, delete this.

underlining the importance of feature extraction and computational modeling in improving detection accuracy. Zhu et al. (2021) [4] investigated the application of the Random Forest algorithm in power systems, noting the efficiency of clustering algorithms and data models essential for detecting intrusions in such complex environments. They discussed the necessity of adapting intrusion detection methods to specific industrial contexts. Jha et al. (2022) [5] proposed an enhanced IDS that combines feature ranking with machine learning algorithms. Their approach correlated system performance with network security, utilizing feature extraction to improve the efficiency of Random Forests in intrusion detection. Their study provided valuable insights into the integration of data collection and classification processes to build effective IDSs.

All these Machine learning-based solutions need for extensive feature engineering, as emphasized by Shah et al. (2020) and Jha et al. (2022), which can be a limitation. The process is often labor-intensive and requires domain expertise, which might not be readily available. ML models often require retraining or fine-tuning as network environments evolve or new types of attacks emerge, which can be resource-intensive. Additionally, traditional ML models may not efficiently process time-series data or sequential patterns that are crucial in IDS for detecting sophisticated attack strategies that develop over time.

To Overcome the issues faced by ML models, people used other approaches like anomaly based IDS. Bhadauria Mohanty (2021) [6] Proposed a hybrid intrusion detection system combining signature-based and unsupervised anomaly-based detection using algorithms like Decision Tree, Naive Bayes, DBSCAN, and Isolation Forest. The hybrid model aimed to improve detection rates of both known and novel network attacks while aiming to lower false alarm rates. Despite improvements, the hybrid system still faces challenges with high false-positive rates typical of anomaly-based detection methods. Jha et al. (2024) [7] Utilized the ER-VEC (Extended Representation Vector Encoding) model to detect anomalies in the CICIDS-2017 dataset, focusing on cyber-attacks and anomaly detection. The use of ER-VEC represents an innovative approach to enhance the detection of sophisticated cyber-attacks, leveraging a robust representation of network traffic for better accuracy. The study might be limited by the computational complexity of the ER-VEC approach and its scalability when applied to larger, more diverse datasets outside the controlled settings of CICIDS-2017. Anomaly-based approaches that incorporate some supervised learning features pave the way for models to lean more towards class properties, enabling them to start learning with some prior knowledge. That prior learning is mostly provided my MLP models.

In recent years, significant strides have been made in the field of intrusion detection systems (IDS) utilizing deep learning techniques, aiming to enhance real-time detection capabilities and improve the robustness of security measures against sophisticated cyber threats. Lets see some of the key contributions from several studies, highlighting various deep learning approaches and their impacts on the efficacy of IDS.

One of the notable advancements is the development of real-time network intrusion detection systems that leverage deep learning architectures for adaptive data acquisition and processing. For instance, Dong et al. (2019) [9] introduced a real-time IDS based on self-encoder neural networks, focusing on the system's ability to adaptively acquire and analyze network data. Similarly, Krishna et al. (2020) [12] utilized Multilayer Perceptrons (MLP) to enhance the intrusion detection and prevention capabilities, demonstrating the flexibility of deep learning models in handling diverse intrusion scenarios effectively.

Further contributions include the application of Convolutional Neural Networks (CNNs) to improve feature extraction processes. Chen et al. (2020) [**?**] explored the use of CNNs for extracting significant features from telecommunication traffic, benchmarking their performance against traditional machine learning methods, thereby showcasing the superior capability of CNNs in detecting nuanced patterns associated with network intrusions.

The integration of advanced deep learning techniques such as combining cross-correlation with Deep Neural Networks (DNNs) and Long Short-Term Memory networks (LSTMs) has also been prominent. Singh and Prakash (2024) [13] proposed a hybrid system that enhances the detection efficiency by leveraging multiple neural network techniques to handle complex data patterns. Additionally, Khan et al. (2023) [14] extended the application of deep learning-based IDS to social networking sites, employing behavioral analyses to detect anomalies, thus broadening the scope of IDS applications beyond conventional network settings.

Moreover, the integration of artificial intelligence (AI) and machine learning algorithms into IDS design has been extensively explored. Li (2023) incorporated AI algorithms such as K-means clustering in the design of IDS, utilizing kurtosis to analyze data properties and enhance system defense against sophisticated attacks. Salim and Hasoon (2024) further reviewed various AI techniques including deep learning and meta-learning, emphasizing their potential to significantly improve the predictive accuracy of IDS against cyberattacks.

In conclusion, while these advancements demonstrate the potent capabilities of deep learning in revolutionizing IDS, there remains an essential need for continued focus on feature property analysis. This focus is crucial for improving decision-making processes concerning new data and variant patterns, ultimately enhancing classification accuracy and the reliability of IDS in adapting to evolving network threats. In our framework we try to emphasis more on method to select the best features that contribute in distinguishing the classes. Learning the patterns across the samples and features for reasoning of the decision made by models. This can be achieved by knowing /having the knowledge of statics of the feature space and derive the new intuitions from the attention (transform model) based features. To obtain this we have used TabNet, Attentive Interpretable Tabular Learning [16]. This provide a way to use ML models with statical and deep learning

parameters to perform the decision with in the ML models. As an extension to this work we have also experimented with FT Transforms to encode the reason of decision by models.

In conclusion, these advancements underscore the powerful capabilities of deep learning in transforming Intrusion Detection Systems (IDS). However, there remains a critical need for ongoing focus on feature property analysis. This focus is vital for enhancing decision-making processes related to new data and variant patterns, ultimately improving classification accuracy and the reliability of IDS in adapting to evolving network threats. In our framework, we place significant emphasis on methods to select the best features that contribute in distinguishing the given two classes. We aim to understand the patterns across samples and features to rationalize the decisions made by models. This understanding can be achieved by gaining insights from the statistics of the feature space and deriving new intuitions from attention-based features, such as those provided by transformer models. To accomplish this, we have utilized TabNet, Attentive Interpretable Tabular Learning [16], which allows the integration of learning models with both statistical and deep learning parameters to make informed decisions within the models. As an extension of this work, we have also experimented with TabTransformer [17] to encode the reasoning behind decisions made by models.

III. Methodology

In this section we will see detailed proposed framework to detect the intrusion in the network. The Fig 1 Shows the block diagram view of the proposed framework, when somebody try to use internet through networks, we can collect some of the network information like, IP connection, protocol duration, protocol type, services, some information can be derived from the payload of the packets so on. We collect these network information in the data base. At initial point the data base cant to be zero information thus we use some of the standard Intrusion datasets to learn and experiment. Once we have network information stored in data base it can be fetched and used for learning purpose. The learning block include the deep learning models TabNet and TabTransformer to train the model based on statical richness and attention based features to influence the decision. We also have a feature selection block which is mostly used to construct features in the training phase of the model. Instead of feeding all the features directly to train the model, we pass these through Sequential Feature Selector - technique used in machine learning to select features based on a criterion that measures the performance of a model with respect to the inclusion or exclusion of individual features. This provide use a list of features that are crucial for distinguish between the output classes. The selected features are provided with more focus while performing the training in the TabNet.

A. Datasets

In this work, we have used three datasets, namely NSL-KDD, CIDDS-002 and CIC-IDS-2018. Lets see more about each datasets in details.

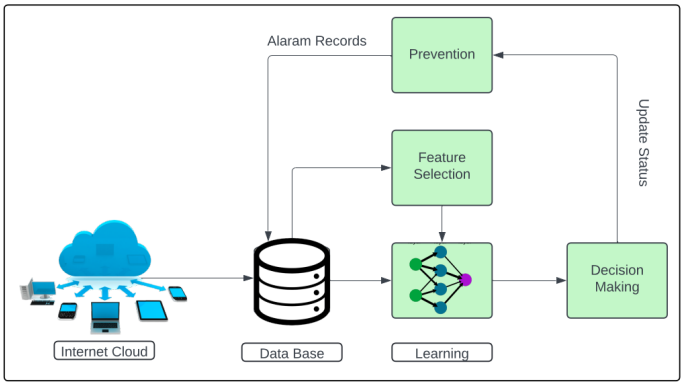

Fig. 1. The Figure shows the block diagram of proposed framework.

1) NSL-KDD: The NSL-KDD dataset is an enhanced version of the KDD'99 dataset, specifically designed to address several of its shortcomings, notably the presence of redundant records and imbalanced class distributions that biased learning and evaluation in intrusion detection models. Furthermore, NSL-KDD provides a testing set that includes new types of attacks, posing a realistic challenge that tests the model's adaptability to novel threats while maintaining a manageable dataset size. This makes it feasible for researchers with limited computational resources to use the entire dataset for training and testing, making NSL-KDD a valuable resource for developing robust and effective intrusion detection systems.

The NSL-KDD dataset comprises of features extracted from network traffic, making it ideal for training and testing intrusion detection systems. The Basic Features are derived from TCP/IP connection packets and include metrics such as the duration of the connection, the type of protocol used (TCP, UDP, ICMP), the network service involved (HTTP, SMTP, etc.), and the status of the connection (normal or error). The Content Features delve deeper into the payload of the packets, identifying critical indicators of malicious activities such as failed login attempts, which are often signs of brute-force attacks. Traffic Features are calculated over a two-second temporal window and provide insights into the network behavior, such as the number of connections to the same host or service. Additionally, each connection in the dataset is labeled as 'normal' or as an 'attack', with the attack types further classified as DOS, Probe, U2R, R2L. The actual dataset has 22 sub classes under attack type. As these classes are compelety skewed, for simplicity we have grouped them into 4 attack types (basically we have 5 classes normal, DoS, Probe, R2L and U2R).

2) CIDDS-002: The CIDDS-002 dataset, part of the Coburg Intrusion Detection Data Sets, is a specialized resource for network intrusion detection, focusing on external traffic analysis. It stands out due to its incorporation of real traffic data, which simulates external attacks on network environments, rather than relying solely on simulated or synthetic data. This approach provides a more realistic testing ground for intrusion detection systems. CIDDS-002 includes compre-

hensive flow-based features such as source and destination IP addresses, port numbers, timestamps, connection durations, the volume of data transferred in bytes, packet counts, and TCP flags. These features enable detailed analysis of traffic patterns and behaviors that are indicative of various types of network attacks. In this dataset after dropping some of the columns we end up using 12 columns as inputs to train the model. The labels are categorizing traffic into 'normal' or as one of several types of attacks, such as port scans, Distributed Denial of Service (DoS) attacks, and SSH brute force attacks. In total we have 44 unique different attacks, we have grouped them into 5 types of attacks. (To simplify the learning process)

3) CIC-IDS-2018: The CIC-IDS-2018 dataset, developed by the Canadian Institute for Cybersecurity, is recognized as one of the most detailed and comprehensive datasets available for intrusion detection studies. It is structured to provide a deep understanding of attack patterns through detailed time-based features, including precise timestamps which are instrumental in analyzing how attack behaviors unfold over time. The dataset contain a rich set of traffic features such as the duration of flows, total number of packets sent and received (forward and backward), the total length of these packets, and the rate of data transfer (flow bytes per second). These features contribute to a granular view of network traffic, allowing for sophisticated analysis and modeling. This dataset in total excluding the label has 78 input features out of which after removing some of features because of high co-rrelation and empty columns we end up getting 68 columns.

Furthermore, the CIC-IDS-2018 dataset encompasses a wide array of attack types, including, but not limited to, Heartbleed, Botnet, Infiltration, Web Attacks, and Distributed Denial of Service (DDoS). Each network flow within the dataset is labeled as either benign or malicious, with the specific type of attack clearly identified. Additionally, the dataset is organized based on records obtained on different days of the week, reflecting the variability in attack occurrence. While training the models we have used two approaches: one that combines data from different weekdays and another that treats weekdays separately.

B. Feature Selection

The learning models decision making is derived by the input features. Especially when the input is in tabular format, we need to very careful in using the features for learning, even small mistake in feature engineer will effect the accuracy of the model. In our framework, we use the Sequential Feature Selector (SFS). Its a feature selection technique used in machine learning to enhance model performance by reducing the dimensionality of the input data. SFS works by either sequentially adding or removing features to form the optimal subset based on a specified performance criterion. This process can be performed in two modes: forward selection and backward elimination. In our work we have employed forward selection, where SFS starts with an empty set of features and adds one feature at a time—the feature that provides the most significant improvement in model performance until no further improvement is observed or a specified number of features (User has to provide this k-features) is reached. We use SFS to know which features dominate to help in distinguishing across different classes. Knowing this will help use to train the learning models in the controlled environment.

C. Learning Method

In our proposed framework, our key object is to address the gap between training the learning models with just statistical analysis (ML Models) and deep learning models learning without knowing what exactly the input data is which make neural model not prone to noise in the input data. There is a need to come up with an approach of learning where we use the knowledge of features statistics to train the model and derive the relation between the features, there changing values impact on the decision using attention model. Models learned based on this technique can be used for reasoning for the provided decision. To achieve this, we employed TabNet [18], a deep learning architecture designed by Google Cloud AI for tabular data. TabNet distinguishes itself with its sequential attention mechanism, which selectively highlights salient features at each decision step. This enhances interpretability and efficiency in learning, making it highly effective for handling the complex, multi-dimensional nature of network traffic data. Thus, we believe that, TabNet is well-suited for intrusion detection systems (IDS). Let's explore in detail how TabNet functions and the specific adaptations we implemented towards supporting it for the IDS applications.

As we mentioned early, the TabNet leverages the sequential attention mechanism to make decisions at each step of the decision process. While selecting the features it selects based on the objective of sparsity using the sparsemax function, which encourages the model to use only the most relevant features at each decision step. It helps in reducing the model's complexity and enhances interpretability by highlighting which features are influential in making predictions. These features are further fed through the feature transformer which is backbone of the TabNet. It providing the basis for both the attentive transformer and the final prediction decision. It is similar to the transformers used in natural language processing but adapted for tabular data. It has the feature transformer which uses multiple layers of transformations, including non-linear processing through gated linear units (GLUs), to encode the input features into a useful representation for making predictions. This kind of technique is anticipated for learning the context between the input features to make model capable of reasoning its features contribution towards the classification.

Before we train the TabNet for classification task, we pass the input train data through TabNet encoder and decoder. This encoder is composed of a feature transformer, an attentive transformer and feature masking. It helps model to learn the feature representation in the better way. Once this step is done, we use the encoder weights to train the classifier. We have experimented with different hyper-parameters across different datasets, more details are presented in the experiment section. While training the classifer as most of our datasets have

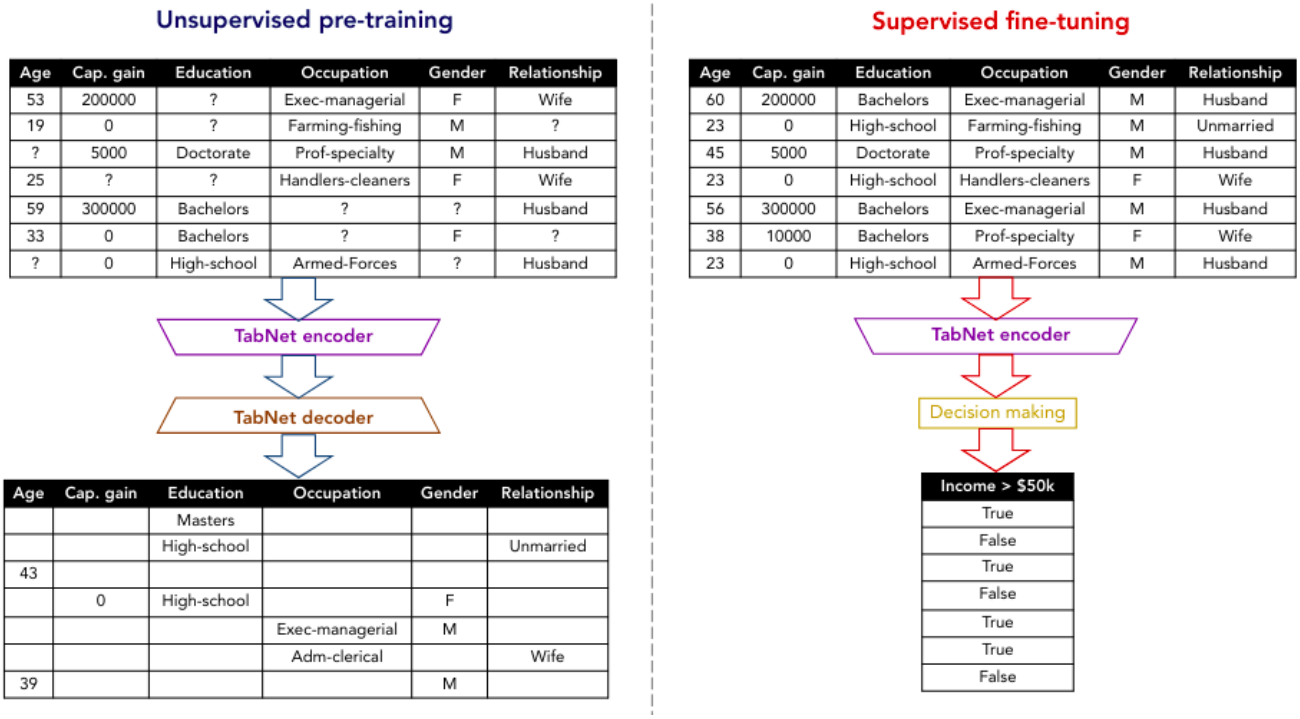

Fig. 2. Self-supervised tabular learning. Showing Both TabNet Encoder and Decoder model and Supervised Classifier - Src Original paper - TabNet

skewed classes, we have employed weighted cross entropy loss function which is explained in section III D.

D. Weighted Cross-Entropy Loss for Multi-Class Classification

As most of our Intrusion detection datasets have skewed classes. We use the weighted cross entropy loss. We calculate weights per class as inversely proportional to their frequency in the dataset. It can be done by first getting the count of samples for each classes. Then calculate the total number of samples and divides this by the count for each class to find their inverse frequency. These obtained weights are normalized by scaling in such a way that the smallest weight would be 1.0, to ensure that no class's weight is disproportionately high. These normalized weights can be used during model training to adjust the influence of each class, particularly helping the model to pay more attention to underrepresented classes and thus addressing class imbalance in training data. For training the model we have used Cross Entropy Loss function along with these weights, below we have equations for calculating weights and loss.

$$w_c = \frac{\sum_{i=1}^{C} n_i}{n_c} \tag{1}$$

where:

- w_c is the weight for class c,
- n_c is the number of samples for class c,
- C is the total number of classes,
- n_i is the number of samples for each class i,
- $\sum_{i=1}^{C} n_i$ is the total number of samples across all classes.

To normalize these weights such that the smallest weight is 1, we apply:

$$w_c = \frac{w_c}{\min(w)} \tag{2}$$

The weighted cross-entropy loss for a single prediction and true label can be defined as:

$$L_{\text{CE}}(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbf{y}_{ic} w_c \log \left(\frac{e^{\hat{\mathbf{y}}_{ic}}}{\sum_{k=1}^{C} e^{\hat{\mathbf{y}}_{ik}}} \right)$$

Here, C is the number of classes, N is the number of samples, $\mathbf{y}_{ic}$ is the target label for instance i and class c, w_c is the weight to penalize the loss function to focus on some specific class and $\hat{\mathbf{y}}_{ic}$ are the logits from the model for instance, i and class c.

IV. Experiments and Results

In this section, we will discuss about the experiment setup and its outcomes across multiple Machine learning and deep learning models. To demonstrate our work we have used 3 different datasets namely NSL-KDD, CIDDS-002 and CIC-IDS-2018. In our experimentation, we tailored the TabNetClassifier model to address the unique challenges of intrusion detection tasks. Through careful hyperparameter tuning, we optimized the decision dimension (n_d) to 64, aiming to enhance the model's capacity to capture complex patterns inherent in intrusion datasets. While keeping the attention dimension (n_a)

at its default value, we leveraged the TabNet architecture's inherent adaptability to focus on relevant features crucial for intrusion detection. We chose Adam optimizer, recognizing its potential to navigate large, complex datasets effectively. Additionally, we employed the 'sparsemax' mask type, which is well-suited for classification tasks, and utilized the default scheduler parameters, allowing the model to dynamically adjust its learning rate. These adjustments were made with careful consideration of the intricacies of intrusion detection tasks, aiming to optimize the model's performance in identifying and classifying anomalous network activities. We utilized our feature selector to identify the top k features essential for learning. Following the preprocessing step, we trained the TabNet model for intrusion classification tasks.

Table I presents the model performance metrics across various machine learning and deep learning models for NSL-KDD dataset. In the last row, we showcase the results obtained using the modified loss function of TabNet. The crucial aspect here is to differentiate the data using both statistical and attention-based approaches to improve the F1 score for smaller-sized classes. From Table I, we observe an accuracy of 82.35 % for TabNet, with a slightly higher recall (M) achieved. Understanding the data in Table I II III means grasping how different classification models performed on the NSL-KDD dataset, which is used for intrusion detection with five classes. Accuracy gives us a general idea of how well each model classified instances correctly. Precision (W) tells us, on average, how good the models were at spotting relevant instances while avoiding false alarms. Recall (W) shows how well the models managed to catch all relevant instances while minimizing misses. F1 (W) combines precision and recall, giving us a balanced view of a model's performance. Precision (M), recall (M), and F1 (M) focus specifically on how well the models handled the minority class, which is crucial in intrusion detection where spotting rare instances is key. By looking at these metrics together, we can figure out where each model excels and where it falls short in terms of accurate classification.

While experimenting with the CiC-IDS-2018 dataset, which comprises 7 GB of data spanning network traffic details over 10 days, we encountered a total of 14 distinct class labels. Our experimentation involved training separate models for each of the 10 records. Remarkably, 8 of these records exhibited an impressive accuracy rate of 99 %. However, the remaining 2 records, tasked with distinguishing between benign versus bot and benign versus malicious (with the exact attack type unspecified), presented significant challenges, leading to sub-optimal learning outcomes. In Figure 3, we depict the input data projected onto a 2D space using t-SNE to visualize the distribution of the two class types within the CIS (benign and bot). It is evident from the visualization that the input features for both types of data lack distinct characteristics, posing challenges for learning models. In such scenarios, employing ensemble-based models becomes imperative to enhance the overall accuracy of the model.

In our paper, we leverage TabNet, to address the challenges

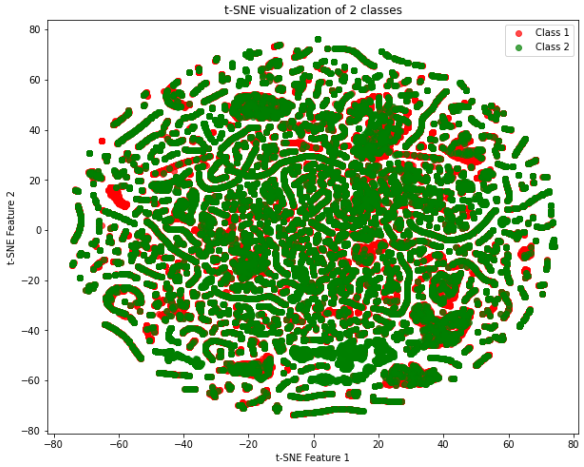

Fig. 3. Showing the 2 labels benign vs bot input data projection on to 2D from the CIC-IDS-2018 dataset.

of feature importance extraction and understanding the source of output activation in the context of intrusion detection using the NSL-KDD & CiC-IDS-2018 datasets. TabNet allows us to extract important features from the input data, providing insights into which features contribute most significantly to the model's decision-making process. Additionally, we focus on understanding the source of output activation in TabNet, shedding light on how the model makes decisions based on input features and learned representations.

To illustrate the impact of single sample inputs on detection, we present a graph in Figure 4, 5 that visualizes how individual instances influence the model's decision-making process. This graph provides a valuable insight into the model's sensitivity to different input samples and helps elucidate the factors driving its predictions. By analyzing these visualizations, computer security engineers may gain clearer insights into how these models are making decisions. If any decisions are found to be incorrect, engineers are provided with a potential pathway to understand the cause.

Overall, our approach combines advanced neural network techniques with interpretability tools to enhance our understanding of intrusion detection systems. By elucidating the importance of features and the source of output activation in TabNet, we provide valuable insights that can inform future research and development in cybersecurity.

The t-SNE based graphs (Figure 6 and Figure 7) illustrate the projection of data onto a 2D space, offering valuable insights into the learned embeddings from TabNet and the distribution of input test data. These visualizations reveal that the TabNet model often generates common probabilistic clusters shared among samples from different classes. While this suggests TabNet effectively captures underlying data patterns, there remains room for improvement in class separation.

A significant observation is the presence of overlapping clusters, indicating ambiguity in learned representations and potential misclassifications. To address this challenge and enhance the model's discriminative ability, we propose em-

Model	Accuracy	Precision (W)	Recall (W)	F1 (W)	Precision (M)	Recall (M)	F1 (M)
Random Forest	81.3%	0.84	0.81	0.79	0.83	0.66	0.68
Decision Tree	78.57%	0.85	0.79	0.78	0.67	0.72	0.61
Gradient Boost	81.71%	0.84	0.82	0.79	0.72	0.60	0.60
AdaBoost	74.17%	0.78	0.74	0.71	0.74	0.51	0.52
SVM	79.08%	0.83	0.79	0.78	0.77	0.7	0.69
CNN1D	80.69%	0.82	0.81	0.80	0.64	0.58	0.59
TabNet	**82.35**%	0.83	0.81	0.81	0.76	0.72	0.69

TABLE I

PERFORMANCE METRICS OF DIFFERENT CLASSIFICATION MODELS TRAINED ON DATASET - NSL-KDD AS A 5 CLASS CLASSIFIER

Model	Accuracy	Precision (W)	Recall (W)	F1 (W)	Precision (M)	Recall (M)	F1 (M)
Random Forest	99.96%	1	1	1	**0.98**	**0.88**	**0.92**
Decision Tree	99.51%	1	1	1	0.61	0.95	0.67
Gradient Boost	99.98%	1	1	1	0.9	0.60	0.66
AdaBoost	99.94%	1	1	1	0.90	0.89	0.89
SVM	99.92%	1	1	1	0.9	0.78	0.83
TabNet	99.92%	1	1	1	**0.88**	**0.84**	**0.86**

TABLE II

PERFORMANCE METRICS OF DIFFERENT CLASSIFICATION MODELS TRAINED ON DATASET - CIDDS-002 AS A 2 CLASS CLASSIFIER

Model	Accuracy	Precision (W)	Recall (W)	F1 (W)	Precision (M)	Recall (M)	F1 (M)
CNN1D	74.0%	0.74	0.76	0.73	0.71	0.63	0.64
TabNet	**76.95**%	0.75	0.77	0.74	0.75	0.72	0.74

TABLE III

PERFORMANCE METRICS OF DIFFERENT CLASSIFICATION MODELS TRAINED ON DATASET - CIC-IDS-2018 (WE ARE JUST SHOWING THE PERFORMANCE OF MODEL WHEN WE CATEGORIZE TWO CLASSES BENIGN VS BOT WHERE DATA IS HIGHLY CORRELATED).

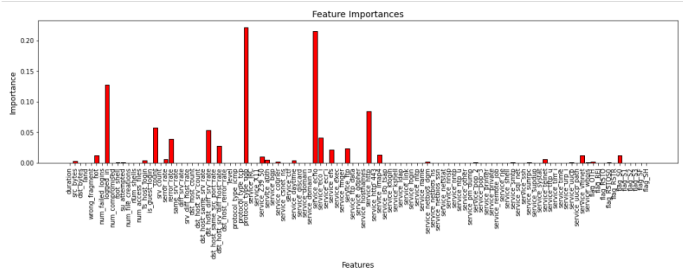

Fig. 4. Showing the NSL-KDD dataset features importance while deciding the output classes. The features are obtained by the TabNet with modified weighted loss model

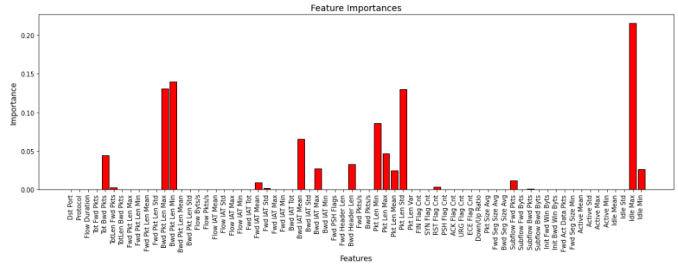

Fig. 5. Showing the CIC-IDS-2018 dataset features importance while deciding between . These features are obtained by the TabNet with modified weighted loss model

ploying an actual transformer model instead of TabNet alone. Transformer models, renowned for their success in natural language processing and image recognition tasks, offer powerful feature extraction capabilities and improved class separation.

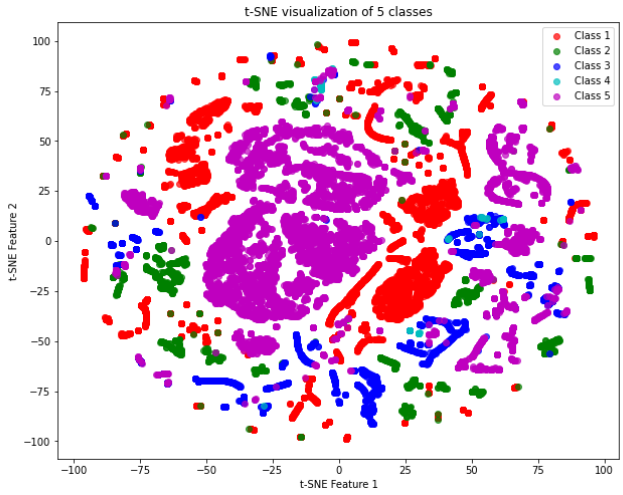

Fig. 6. tSNE based graph showing the projection of input test data on the 2D space.

By integrating transformer models, the model can leverage pre-existing knowledge and learn more meaningful and discriminative representations. This approach has the potential to enhance the model's performance in distinguishing between different classes and reducing misclassifications. Addditionlly we can also provide verbal reasoning for the decision made by the model. Overall, this insight underscores the importance of

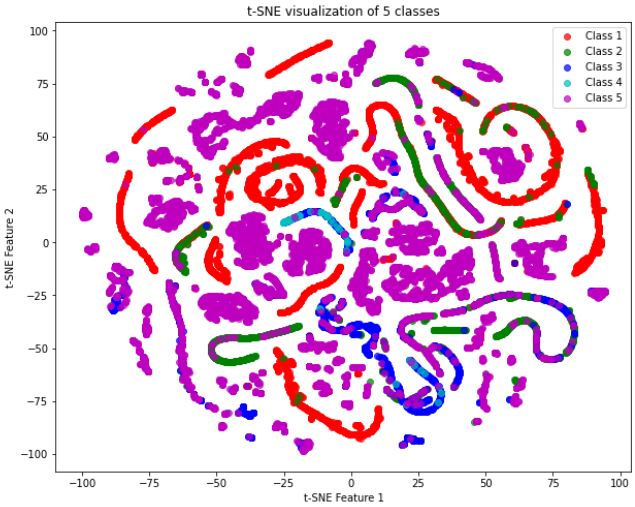

Fig. 7. tSNE based graph showing the projection of learned embeddings from TabNet on the 2D space.

exploring alternative modeling approaches to further advance the effectiveness of intrusion detection systems.

V. Conclusion

In this paper, we disccuss about the leaning models that are used in Intrusion Detection System. We use TabNet as a base model in our work to make classification decisions involving both statical and attention based featires. As demonstrated in our research. However, the need for ongoing focus on feature property analysis remains critical. Our framework emphasizes selecting the best features that contribute to distinguishing between classes and understanding the patterns across samples and features to rationalize decisions made by models. This is achieved by leveraging statistical insights from the feature space and deriving new intuitions from attention-based features from the TabNet. Even though we try to provide a reasoning for the detected Intrusion but we were not able to get improved accuracy by this model. Our future work will focus on refining these techniques and exploring additional deep learning strategies especially the TabTransformers - Transformers used for tabular data (Mostly these are used for categorical features) to further improve the IDS's adaptability to evolving network threats.

References

[1] S. Kumar, S. Gupta and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," in IEEE Access, vol. 9, pp. 157761-157779, 2021, doi: 10.1109/ACCESS.2021.3129775. keywords: Intrusion detection;Market research;Computer security;Search engines;Feature extraction;Computer hacking;Machine learning;Citation;machine learning;bio-inspired;intrusion detection system;NIDS;datasets,

[2] A. Kiran, S. W. Prakash, B. A. Kumar, Likhitha, T. Sameeratmaja and U. S. S. R. Charan, "Intrusion Detection System Using Machine Learning," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-4, doi: 10.1109/ICCCI56745.2023.10128363. keywords: Computers;Support vector machines;Intrusion detection;Network intrusion detection;Machine learning;Software;Hardware;Support vector machine;Machine Learning;Network Intrusion Detection System;Host Intrusion Detection System;Intrusion Prevention System;Intrusion Detection System;Host;Network,

[3] A. Shah, S. Clachar, M. Minimair and D. Cook, "Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems," 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 2020, pp. 759-760, doi: 10.1109/DSAA49011.2020.00102. keywords: Telecommunication traffic;Computer science;Feature extraction;Computational modeling;Network intrusion detection;Measurement;Biological neural networks;Network Intrusion Detection System;Signature-based Intrusion Detection System;Anomaly-based intrusion detection system;multiclass classification,

[4] G. ZHU, H. YUAN, Y. ZHUANG, Y. GUO, X. ZHANG and S. QIU, "Research on network intrusion detection method of power system based on random forest algorithm," 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 2021, pp. 374-379, doi: 10.1109/ICMTMA52658.2021.00087. keywords: Power measurement;Mechatronics;Image edge detection;Network intrusion detection;Clustering algorithms;Data models;Power systems;Random forest algorithm;Power system;Network intrusion;Intrusion detection,

[5] Raman, S. K. Jha and A. Arora, "An Enhanced Intrusion Detection System Using Combinational Feature Ranking and Machine Learning Algorithms," 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 2022, pp. 1-8, doi: 10.1109/CONIT55038.2022.9847815. keywords: Machine learning algorithms;Correlation;System performance;Intrusion detection;Network security;Feature extraction;Random forests;Intrusion Detection Systems(IDSs);Data collection;Feature ranking;Classification,

[6] S. Bhadauria and T. Mohanty, "Hybrid Intrusion Detection System using an Unsupervised method for Anomaly-based Detection," 2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Hyderabad, India, 2021, pp. 1-6, doi: 10.1109/ANTS52808.2021.9936919. keywords: Network intrusion detection;Clustering algorithms;Forestry;Feature extraction;Chatbots;Telecommunications;Safety;network intrusion detection system;unsupervised;signature-based attack;anomaly-based attack,

[7] R. S. Jha, K. Ojha, A. Mishra, R. Mishra and A. Kaushik, "Cyber-Attacks and Anomaly detection on CICIDS-2017 dataset using ER-VEC," 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024, pp. 1453-1458, doi: 10.1109/ICDT61202.2024.10489209. keywords: Technological innovation;Firewalls (computing);Data security;Intrusion detection;Predictive models;Vectors;Ransomware;firewall;anomaly;detection;cybersecurity;cyber-attack IDS;IPS;DT;RF;EXTC;ER-VEC,

[8] Ainurrochman et al., "Ensemble Methods Classifier Comparison for Anomaly Based Intrusion Detection System on CIDDS-002 Dataset," 2021 13th International Conference on Information Communication Technology and System (ICTS), Surabaya, Indonesia, 2021, pp. 62-67, doi: 10.1109/ICTS52701.2021.9608714. keywords: Training;Statistical analysis;Stacking;Intrusion detection;Gaussian processes;Decision trees;Security;Ensemble Methods;Anomaly-Based Intrusion Detection System;CIDDS;Accuracy;Detection Rate,

[9] Y. Dong, R. Wang and J. He, "Real-Time Network Intrusion Detection System Based on Deep Learning," 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2019, pp. 1-4, doi: 10.1109/ICSESS47205.2019.9040718. keywords: Intrusion detection;Real-time systems;Deep learning;Feature extraction;Anomaly detection;Neural networks;intrusion detection;deep learning;self-encoder;adaptive acquisition,

[10] L. Chen, X. Kuang, A. Xu, S. Suo and Y. Yang, "A Novel Network Intrusion Detection System Based on CNN," 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, 2020, pp. 243-247, doi: 10.1109/CBD51900.2020.00051. keywords: Support vector machines;Network intrusion detection;Telecommunication traffic;Machine learning;Benchmark testing;Feature extraction;Data models;Network Intrusion Detection System;CNN;Deep Learning,

[11] S. Li, "Design of Network Intrusion Detection and Prevention System Incorporating Artificial Intelligence Algorithms," 2023 5th International Conference on Applied Machine Learning (ICAML), Dalian, China, 2023, pp. 581-587, doi: 10.1109/ICAML60083.2023.00113. keywords: Deep learning;Machine learning algorithms;Fluctuations;Network intru-

sion detection;Network security;Kurtosis;Data models;K-means algorithm;Kurtosis density value;Intrusion detection;Defense;System design,

[12] A. Krishna, A. Lal M.A., A. J. Mathewkutty, D. S. Jacob and M. Hari, "Intrusion Detection and Prevention System Using Deep Learning," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 273-278, doi: 10.1109/ICESC48915.2020.9155711. keywords: Machine learning;Feature extraction;Intrusion detection;Data models;Software;Training;Testing;Deep Learning;Intrusion Detection System;MLP;Prevention;Network,

[13] N. Singh and J. Prakash, "A Hybrid System for Integrating Cross-Correlation, DNNs and LSTMs for Enhanced Intrusion Detection System," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1291-1296, doi: 10.1109/IC2PCT60090.2024.10486243. keywords: Wireless communication;Wireless sensor networks;Time series analysis;Intrusion detection;Telecommunication traffic;Feature extraction;Time measurement;Wireless Sensor Network;DNN;LSTM;Cross-correction;Intrusion detection;Deep learning,

[14] S. S. Khan, A. Deo, K. K. Baraskar, A. Patel, A. Pathak and A. K. Joshi, "Deep Learning Based IDS to Detect Anomaly Over Social Networking Site: Comprehensive Review," 2023 International Conference on Integration of Computational Intelligent System (ICICIS), Pune, India, 2023, pp. 1-7, doi: 10.1109/ICICIS56802.2023.10430316. keywords: Deep learning;Social networking (online);Intrusion detection;Computer architecture;Feature extraction;Behavioral sciences;Anomaly detection;OSN;IDS;security attacks;behaviours;patterns;soft computing;deep learning;machine learning,

[15] Z. H. Salim and S. O. Hasoon, "Intrusion Detection Using Artificial Intelligence Techniques: Review," 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Victoria, Seychelles, 2024, pp. 1-7, doi: 10.1109/ACDSA59508.2024.10467524. keywords: Metalearning;Deep learning;Adaptation models;Machine learning algorithms;Reviews;Intrusion detection;Predictive models;Intrusion Detection Systems (IDS);Deep Learning;Meta-Learning;cyberattacks

[16] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning", AAAI, vol. 35, no. 8, pp. 6679-6687, May 2021.

[17] Xin Huang, Ashish Khetan, Milan Cvitkovic and Zohar S. Karnin, Tabular Data Modeling Using Contextual Embeddings, CoRR, abs/2012.06678, 2020.

[18] Sercan Ömer Arik and Tomas Pfister, "TabNet: Attentive Interpretable Tabular Learning", CoRR, abs/1908.07442, year 2019.