Computer Engineering and Applications Vol. 15, No. 2, June 2026

Hybrid Interpretable and Deep Learning Models for Intrusion Detection in Large-Scale Network Traffic

Chintureena Thingom[1], M K Harikeerthan[2], S Cloudin[3], K Lokeshwaran[4], K.Praveena[5], K.R. Prasanna Kumar[6], P. Deepa[7], Kishore Chandra Dev Nakka[8]

[1]*Department of Computer Science and Engineering, Aditya University, Surampalem, Andhra Pradesh 533437, India.*
[2]*Department of Civil Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka 560082, India.*
[3]*Department of Computer Science and Engineering, Easwari Engineering College, Chennai, Tamil Nadu 600089, India.*
[4]*Department of Computer Science and Engineering, School of Computing, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu 600062, India.*
[5]*Department of Electronics and Communication Engineering, School of Engineering, Mohan Babu University, Tirupati, Andhra Pradesh 517102, India.*
[6]*Department of Computer Science and Design, Kongu Engineering College, Erode, Tamil Nadu 638060, India.*
[7]*Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamil Nadu 600123, India.*
[8]*Department of Electronics and Communication Engineering, S R K R Engineering College, Bhimavaram, Andhra Pradesh 534202, India.*
* *meera0010001@gmail.com*

ABSTRACT

The fast growth of cyber-attacks and network traffic, have put forward the requirement of autonomous and scalable IDSs that can accurately discern among normal and malicious activities. In this paper, a hybrid machine learning (ML)-based IDS model, DTCNN-IDS, is presented by combining Decision Tree (DT), Convolutional Neural Network (CNN), and TabTransformer. The framework is tested against the KDD99 data set, containing 4,898,431 network records with continuous and categorical fields. A uniform pipeline with preprocessing, encoding, normalization, and multi-class supervised learning (M2A approach) allows for robust model evaluation. DT produces high accuracy (99.99%) but biased results on minority attacks (U2R recall = 0.72, R2L recall = 0.76) as a result of class imbalance. CNN enhances the nonlinear feature learning and achieves an accuracy of 99.7% with the precision, recall and F1-score of 0.996. The best-performing model is TabTransformer, achieving accuracy of 99.8%, precision of 0.997, recall of 0.998 and F1-score of 0.997, which also significantly improves detection of minority attacks. The improved sensitivity and stability are further confirmed by the Precision–Recall, scalability analyses and statistical testing ($p < 0.05$) validates the significance of results.

Keywords: Intrusion Detection, Decision Tree, Convolutional Neural Network, TabTransformer, Large-Scale Network Traffic.

1. INTRODUCTION

The proliferation of digital communication networks and the exponential growth of internet-connected devices have resulted in a massive increase in volume and

complexity of network traffic within today's cyber infrastructures. Network security has become a big problem with the popularity of cloud computing, e-commerce sites, online banking systems, and smart connected services. Intrusion Detection System (IDS) are crucial in defending these infrastructures by detecting malicious activities such as denial of service (DoS), probing, user to root (U2R), and remote to local (R2L) attacks. Recent studies confirm that worldwide cyberattacks are escalating in numbers and sophistication, emphasizing the urgent demand for intelligent and flexible intrusion detection systems capable of processing high-volume data streams from network flow in real time.

The conventional signature-based IDS methods rely heavily on predefined patterns of attack. Thus, they have difficulty in detecting unknown or evolving cyber attacks. Here, we show that machine learning-based intrusion detection is a promising technique. It is able to automatically learn sophisticated patterns from historical network traffic data [24]. Using statistical learning theory and pattern recognition, these data-driven methods enable a clean separation between normal and malicious behaviors, without the need of any rule definitions. However, the performance of traditional shallow learning models are still limited by high dimensionality, heterogeneity of feature types, severe class imbalance, and nonlinear attack patterns.

The KDD99 dataset has been the standard benchmark to test IDS for many years since it is representative of normal traffic as well as attacks of multiple categories. The dataset has billions of labeled network connection records. These are numerical and categorical. They include connection-level features, that is, duration, protocol type, service type, number of bytes sent, and host-level features such as error rates and number of connections.

These features incidentally also provide good context for intrusion behavior modeling. They are, however, more complicated due to high variance and class imbalanced attack distributions, skewed towards DoS attacks, weighted mainly. Consequently, an efficient IDS should take into consideration various features, be capable of learning nonlinear decision boundaries, and be able to detect rare attack types at high detection rates such as U2R and R2L.

The initial intrusion detection methods focused on classical machine learning algorithms (decision tree, support vector machine, rule-based classifiers). Decision tree models also have the advantage of being interpretable and relatively fast to execute. They allow security analysts to understand how classifications are performed based on rules derived from network traffic features. However, these models have a tendency to overfit and do not generalize well when dealing with large and skewed datasets. With the increasing complexity of cyber threats, the need for deep learning models which can learn both hierarchical and contextual feature representations have become more evident.

Deep learning architectures, in particular Convolutional Neural Networks (CNNs), have demonstrated great promise for intrusion detection since they can automatically extract spatial and local feature relations from high-dimensional traffic data. CNNs transform raw feature vectors into hierarchical representations through convolutional filters. This enables the identification of more complex attack patterns, which are difficult to be detected by traditional rule-based approaches. But CNNs are

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

oriented towards local feature interactions and they may not be the best tool to model long-range dependencies that may exist between features in network intrusion datasets.

To address this challenge, transformer-based models have been developed for tabular data analysis in recent years. The TabTransformer architecture exploits the multi-head self-attention mechanism to capture the contextual information among categorical and numerical features. This enables the model to pay more attention to informative features in classification. This attention-based learning strategy not only enhances generalization, but also brings substantial improvements in discovering rarely seen and/or sneaky attacks by exploiting global feature relationships which are hard to be grasped by traditional deep models.

Pradeepthi and Maheswari (2024) [15] proposed an intrusion detection and prevention mechanism that incorporates detection with data encryption to enhance end-to-end security. Their work shows that just detecting attacks is not enough; there is a need to protect the transmitted data during or after detection. Their hybrid classifier methodology demonstrates that multi-stage IDS architectures can achieve improved resilience by combining classification methodologies with other BS functionalities. This work also highlights the need for holistic security systems, in which detection accuracy needs to be balanced with mission-centric security considerations such as secrecy and readiness to respond.

Kamal and Mashaly (2025) [16] introduced a hybrid IDS pipeline based on Principal Component Analysis (PCA) with a Transformer-based, mainly emphasizing on learning effectively from merged datasets. Their result provides evidence that transformers can extract meaningful representations from traffic features when dimensionality is reduced via feature reduction. This paper is significant for large-scale intrusion detection as two major implications are (i) transformer architectures can achieve better generalization for complex distributions and (ii) the refined feature space (e.g., PCA) can reduce redundancy and boost convergence. This encourages us to add TabTransformer, which can handle heterogeneous tabular data, to modern IDS pipelines.

Yassine et al. (2025) [17] the authors discuss a two-level centralized tree-based IDS with decidable layered detection. Their method demonstrates that decision-tree families are very good candidates for IDS since they exhibit a very high level of interpretability, very low latency and the possibility to extract rules that can help in forensic analysis. The multi-level approach reveals, the way in which a tree refinement level could refine both awareness and classification reliability. This study contributes to the mandate of tree-based baselines in IDS research and advocates the use of DT as a crisp baseline architecture in comparative analysis.

Momand et al. (2024) [18] developed ABCNN-IDS which integrates attention mechanism with CNNs for intrusion detection over IoT. Their findings bolster the argument that CNN effectively learns network traffic features patterns and attention enhances the performance by focusing on important feature regions. This is in line with the requirements from the current IDSs when the raw data have complicated correlation and imbalanced distribution. Their findings show that CNN-based

architectures constitute strong deep baselines and that attention-enhanced deep learning further improves detection of subtle and minority-class attacks.

Sajid et al.(2024) [19] proposed machine learning based deep learning technique for the enhancements detection. It is demonstrated in this study that no single learning approach is dominant in all cases. Traditional ML models are interpretable and computationally efficient, while deep networks can implicitly capture nonlinear interactions among features and higher order complex behaviors. Their results call for unified frameworks that allow different model types to be compared or potentially integrated within a single system. This aligns well with employing a multi-model framework (Decision Tree + CNN + Transformer) as a trade off for completeness, scalability and detectability.

Hu et al. (2025) [20] addressed concept drift, a critical challenge for real-world IDS considering traffic patterns evolve over time due to changing user behaviors, new applications, and advanced attacks. Their double adaptive window approach enables the online learning algorithms to adapt their detection in real-time and therefore improve the robustness against non-stationary environments. This study points out that a high precision on static datasets does not guarantee long-term dependability. So it promotes the extension of IDS systems with drift-aware training and testing. It underpins the argument that models with superior representation learning, e.g., transformers, as base models, have the further potential to generalise better when combined with online update approaches.

Bamber et al. (2025) [21] proposed a combined CNN-LSTM IDS. In this model, the spatial patterns are learnt through the CNN layers and temporal correlations are captured in the LSTM layers. Their method stresses an interesting insight about IDS problem, that attacks may emerge not only in magnitudes of features but also in sequential behaviors or evolving connection states. Although the KDD99 is generally considered as tabular per-connection information, this work justifies that the deep hybrids do capture other aspects like local feature interactions as well as longer behavioural sequences for improved detection. This motivates the utilization of CNN-based modules, and further leads to a unification of transformers with temporal modelling in our future work.

Qiu et al. (2025) [22] proposed to apply deep reinforcement learning (DRL) for optimizing a hybrid IDS architecture comprising CNN and Decision Tree modules. Their experimental results demonstrate that model selection, hyperparameter optimization, and the composition of hybrid architecture can be viewed as a single unified optimization problem for improving the performance of their IDS. This contribution lays the groundwork for hybrid IDSs in which deep models capture complex representations, while tree-based logic enhances interpretability and consistency of decisions. It allows architectures of IDS pipelines to be designed to trade off accuracy for explainability and facilitates the architectural configurations to be optimized more systematically.

Mohammed et al. (2025) [23] proposed a two-stage hybrid Intrusion Detection System (IDS) that captures the Falsified Data Injection (FDI) attack in smart grid scenario. They demonstrate that solutions are often, but not always, tailored to their domain. Smart grids have different signatures for attacks than those of enterprise networks or IoT ones. The work stresses that evaluation must be performed with the

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

context of attack. IDS robust pipelines require adaptable modeling techniques. This justifies the employment of transformer-based models capable of capturing high-order interactions among features. It also instigates the experimentation with IDS schemes in different datasets and domains rather than just KDD99.

Abiramasundari and Ramaswamy (2025) [24] investigated the transformer-based language modeling for detection of ransomware phishing e-mail attack and text-based intrusion vectors. Although this is not related to network flow classification (NFC) but e-mail security, it shows how we can extend our tf-avd transformer to more general-purpose cybersecurity detection. It shows that attention can capture contextual patterns other than those captured by traditional methods. This aligns with the potential of using transformers for network IDS, since the contextual feature (i.e., protocol, service, flag relations) relations themselves are the most important.

Neto et al. (2024) [25] presented a survey of ML-based IoT security in healthcare with an emphasis on datasets. They highlighted the following persistent issues for IoT security datasets: skews, stale attack definition, homogeneity, and noisy labeling. This review is important, as it demonstrates the influence of the dataset quality on the validity and the generalization of IDSs. It points out the necessity of rigorous data processing, strong evaluation metrics other than accuracy, cross-validation among multiple datasets is desirable to examine applicability of developed results to the real-world application. It also specifies that more up-to-date datasets other than KDD99 should be used for testing.

Sasi et al. (2024) [26] proposed scalable Selfgattention-based 1D-CNN-LSTM model for detection and identification of IoT attacks in network traffic. Their findings indicate that attention mechanisms enable deep models to better attend to salient features of traffic, while CNN and LSTM layers complement one another in capturing spatial and temporal features. This corroborates two high-level design principles: (1) a novel attention mechanism can improve the distinguishability under complicated traffic scenarios, and (2) Deep hybrid architectures are superior to single-model techniques in terms of performance. Their findings provide compelling evidence for the effectiveness of using attention-based models, such as TabTransformer, in IDS pipelines and highlight exciting possibilities for the fusion of CNN and attention modules. A thorough examination reveals three recurrent themes that cut across the literature:

1. The combination-based approach contributes to enhancing the robustness of the IDS. It has been found that classical ML and DL approach should be combined to trade-off interpretability and performance ([15], [19], [21], [22]).
2. Attention and transformers enhance the performance in minority and complex attack detection. Transformer- or attention-based methods achieve better context modeling and generalization ([16], [18], [24], [26]).
3. Static accuracy is not enough for the real world IDS. Strong evaluation and up to date datasets are needed drifts in concepts and limitations of datasets ([20], [25]).

Based on these observations, this work presents a common intrusion detection framework. It investigates Decision Tree (interpretable baseline), CNN (deep

hierarchical representation learning) and TabTransformer (contextual attention-based modeling) on KDD99. It highlights imbalanced class behavior and minority attack detection for a major assessment focus.

2. MATERIAL AND METHODS

In this part, the technique of intrusion detection with the KDD99 dataset are introduced. It consists of three models: Decision Tree (DT), Convolutional Neural Network (CNN), and TabTransformer. The method guarantees a machine learning pipeline that integrates raw data preprocessing, feature engineering, conventional rule-based learning, deep learning for convolutional representations, and modeling for feature interactions with transformers.

The purpose of the framework is to develop a scalable and efficient IDS (Intrusion Detection System), which applies signature-based detection for host-based intrusions and anomaly-based detection for network-based intrusions, capable of accurately classifying the network traffic as normal or among different attack types such as DoS, Probe, U2R, and R2L. The methodology comprises system architecture and design, dataset modelling, a preprocessing pipeline, model learning techniques, evaluation protocol, and implementation considerations.

2.1 OVERALL FRAMEWORK ARCHITECTURE

The framework employs a multi-step layered architecture to transform network traffic logs at the raw level into final actionable predictive outputs on intrusion classification.

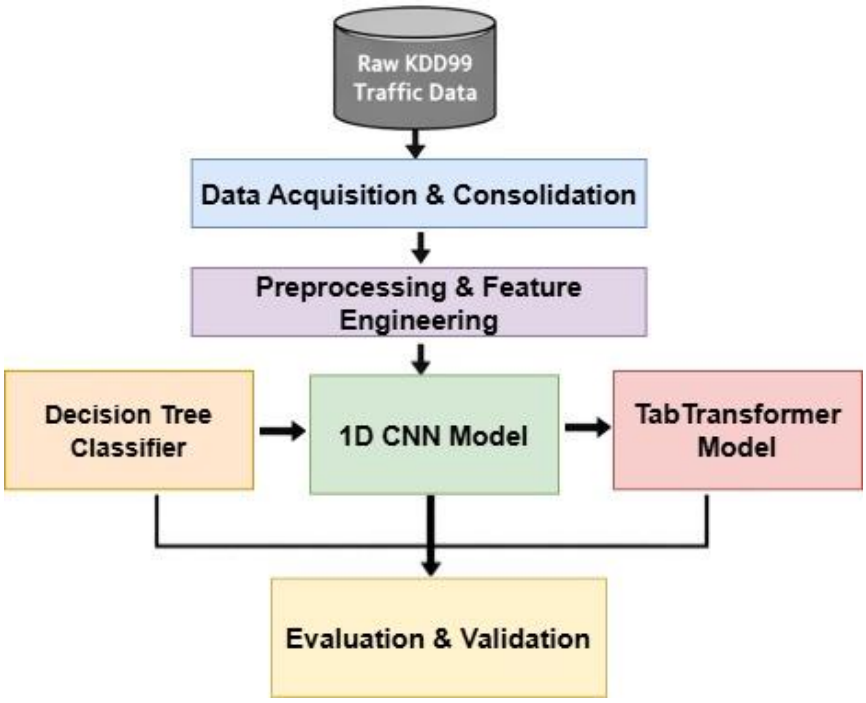

FIGURE 1. Implemented Multi-Model Intrusion Detection Framework Integrating Decision Tree, CNN, and Tabtransformer Models.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

The pipeline in Figure 1 begins with the loading of raw KDD99 traffic records. Then, it includes cleaning, encoding, normalizing and producing feature vectors. The resulting feature space is then fed to three separate classifiers: the entropy-based Decision Tree, 1D CNN, and TabTransformer. This permits comparison and reproducible experiments.

2.2 DATASET MODEL AND ASSUMPTIONS

The KDD99 dataset is considered as a supervised problem for multi-class classification. Each entry corresponds to a network connection and contains numerical as well as categorical fields. Each sample is tagged with a particular type of attack and a general attack category. The approach assumes:

- The attack labels are true and represents real intrusion actions.
- The extracted features are a good representation of the traffic dynamics.
- The training and test sets are statistically representative.
- Supervised deep learning and tree-based methods can be used for multiclass classification.

The modeling parameters and assumptions of the dataset employed in this work are presented in Table 1. It provides an overview of dataset characteristics such as the source of the dataset, the structure of features, the type of attack categories, considerations for preprocessing, and assumptions for model construction and testing.

TABLE 1.
Dataset Modeling Parameters and Assumptions

Parameter	Description
Dataset	KDD99 Intrusion Detection Dataset
Records	4,898,431 network connections
Feature Types	Mixed (Numerical + Categorical)
Target Labels	attack_type and attack_class
Learning Paradigm	Supervised Multiclass Classification
Models Implemented	Decision Tree, CNN, TabTransformer

This design enables fine-grained attack detection and the joint prediction of the intrusion categories in one single IDS.

2.3 DATA ACQUISITION AND CONSOLIDATION

We imported and merged several KDD99 configurations to create a single dataset file. The original files did not have column names headers, so we added standardized attribute names. We defined a mapping function to obtain the attack_type and attack_class labels for hierarchical classification. Data consolidation, schema

standardization, and label mapping in the preprocessing phase of KDD99 is illustrated in figure 2. Several subsets of datasets are merged into a single dataset. The column headers are standardized, and a mapping process creates the labels attack_type and attack_class. This method provides a uniform data format and appropriate clustering of intrusion classes to learn the model.

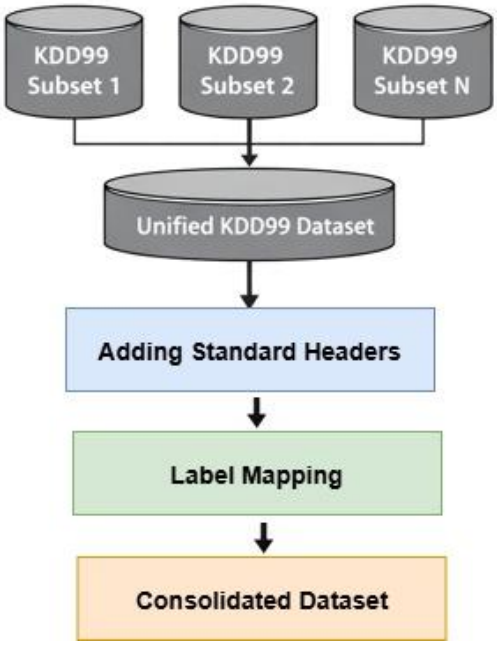

FIGURE 2. Implemented Workflow For Data Consolidation, Schema Standardization, and Label Mapping In KDD99 Preprocessing.

This procedure guaranteed a uniform adjustment of schemas, and the consequent combination of attack types into five general categories.

2.4 DATA PREPROCESSING AND FEATURE ENGINEERING

The preprocessing process applied cleaning, encoding, and normalization techniques to raw traffic logs to extract feature vectors that were ready for the model. The detection and correction of missing, void and inconsistent records were made by means of filtering or imputation to be able to rely on a high quality training data. Categorical attributes such as protocol type, service and flag were converted to numerical values. One-hot encoding executed for DT and CNN model. Embedding encoding was also used for TabTransformer to learn semantic relations between categorical values.

In order to provide stable gradient convergence, numerical features were normalized for training CNN and TabTransformer. As the Decision Tree model is scale invariant, it used the unscaled values hence it was not affected by the feature scaling. The data preprocessing for model training with the dataset describes the procedure of data cleaning, header labeling, label conversion, categorical encoding, and normalization. So each of these steps is merely to have complete, uniformed and optimal input to ML models.

Figure 3 depicts the preprocessing procedure that transforms raw network traffic records into normalized feature vectors for multi-view learning. It involves cleaning

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

the data, assigning headers, mapping labels, encoding categorical variables and normalizing. This makes the whole data set order and full ready for DT and CNN and TabTransformer training.

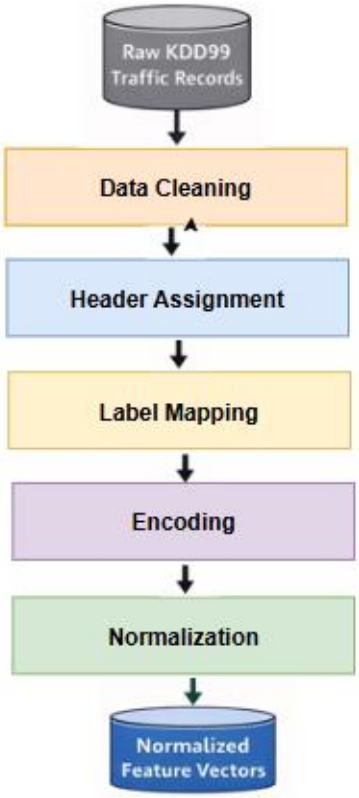

FIGURE 3. Preprocessing Pipeline Transforming Raw Network Traffic Records Into Normalized Feature Vectors for Multi-Model Learning.

2.5 TRAINING AND TESTING PARTITION STRATEGY

The data was divided into a training set and a testing set with a 1:1 ratio in order to provide an unbiased evaluation. A validation subset was also formed from the training data to observe the convergence of the CNN and TabTransformer. The partition scheme of the dataset for learning the model is illustrated in Table 2. In order to achieve sound model training and fair performance assessment, the dataset is divided into training and testing sets.

TABLE 2.
Dataset Partition Configuration

Partition	Purpose	Ratio
Training Set	Model learning	50%
Testing Set	Performance evaluation	50%
Validation Set	Loss monitoring (CNN & Transformer)	Derived from training

This balanced configuration guaranteed that the generalization could be fairly evaluated and the models could be fairly compared.

2.6 DECISION TREE CLASSIFICATION METHOD

The Decision Tree classifier applied the entropy criterion, which is equivalent to information gain, to split features. Recursive partitioning was continued until a maximum depth of 16 was reached in order to avoid overfitting and at the same time to keep the interpretability of the tree. Table 3 shows hyperparameter settings of DT. It contains important parameters including entropy based split criterion, maximum tree depth, random state configuration and the objective of multi-class intrusion classification.

TABLE 3.
Decision Tree Model Configuration

Parameter	Value
Split Criterion	Entropy (Information Gain)
Maximum Depth	16
Random State	Not fixed
Objective	Multiclass intrusion classification

The entropy-based decision tree learning is illustrated in Figure 4 for hierarchical intrusion classification. The information gain criterion of the model splits features recursively to form a tree structure, which partitions the network traffic into different intrusion types according to significant features.

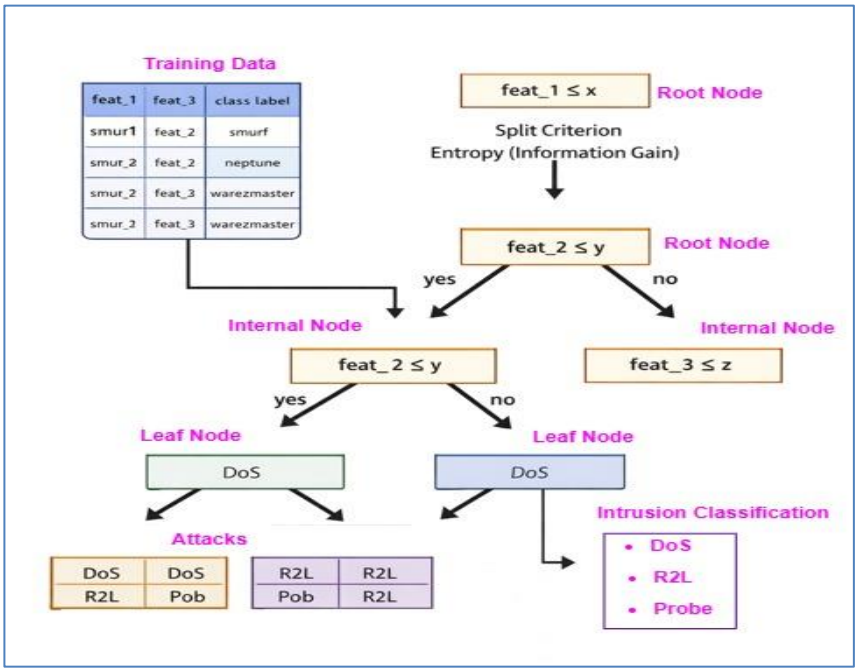

FIGURE 4. Implemented Entropy-Driven Decision Tree Learning Process for Hierarchical Intrusion Classification.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

The resulting model generated interpretable classification rules using features such as dst_host_diff_srv_rate, srv_count, src_bytes. These are powerful signatures of denial-of-service attacks.

2.7 CNN-BASED DEEP LEARNING MODELING

1D Convolutional Neural Network capturing spatial correlation among tabular traffic features. Filters learned local feature interactions directly, and pooling layers furthered dimensionality reduction and improved generalization. Table 4 describes the parameter of CNN model in the evaluation on intrusion detection. Size of the filter in each convolutional layer, the number of activation functions, the optimizer and the number of learning epochs for learning hierarchical feature representation from network traffic flow data is also specified.

TABLE 4.
CNN Architecture Configuration

Layer	Filters / Units	Activation
Input Layer	Feature vector size	—
Conv Layer 1	32 filters	ReLU
Conv Layer 2	64 filters	ReLU
Conv Layer 3	128 filters	ReLU
Dense Layer	128 units	ReLU
Output Layer	Number of classes	Softmax

Figure 5 illustrates the CNN architecture for IDS. The model comprises several convolutional layers to capture the local feature interaction of related network traffic attributes. It is thenpooling and dense layers to learn hierarchical representationsand perform final classification.

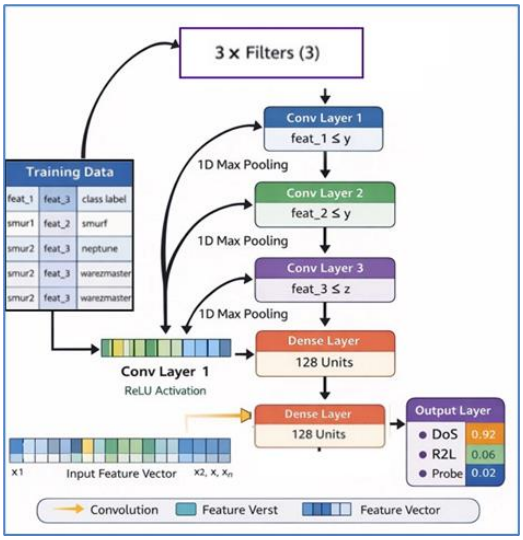

FIGURE 5. Implemented CNN Architecture.

Figure 5 shows convolutional feature extraction and hierarchical representation learning from network traffic data. This CNN model is designed to capture nonlinear relations among traffic features, thereby detecting more complicated rare attacks.

2.8 TABTRANSFORMER-BASED MODELING

The TabTransformer model employs multi-head self-attention to learn contextual relationships between categorical features and numerical features. Categorical features were embedded and processed through stacked transformer encoder layers. Configuration of the TabTransformer for intrusion detection is presented in Table 5. It describes essential hyper-parameters, which are the number of transformer layers, the number of attention heads, the embedding dimension, the number of feedforward units, and finally, the softmax output layer for the multi-class classification.

TABLE 5.
TabTransformer Architecture Configuration

Component	Specification
Transformer Layers	4
Attention Heads	8
Embedding Dimension	32
Feedforward Units	128
Output Layer	Softmax classifier

The model maps categorical features to dense vectors. It leverage multi-head self-attention via stacked transformer layers to capture contextual interactions among features, which leads to better classification of network traffic patterns. The transformer encoder assigned dynamic weights to salient traffic features and thus enhanced the robustness of classifying and detecting minority attacks.

2.9 INTEGRATED IDS PIPELINE WITH MULTI-MODEL LEARNING

The outputs of all three classifiers were combined in a single IDS pipeline, where the preprocessed feature vectors were sequentially fed to the Decision Tree, CNN and TabTransformer. The Decision Tree gave interpretable rule based predictions, the CNN learned deep hierarchical representations and the TabTransformer made predictions based on attention and context. Figure 6 illustrates the combined intrusion detection flow that integrates the DT, CNN, and TabTransformer models.All three classifiers receive the preprocessed feature vectors. Each model predicts with rule-based learning, convolutional feature extraction, and attention-based contextual learning. This can be extremely useful for intrusion detection.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

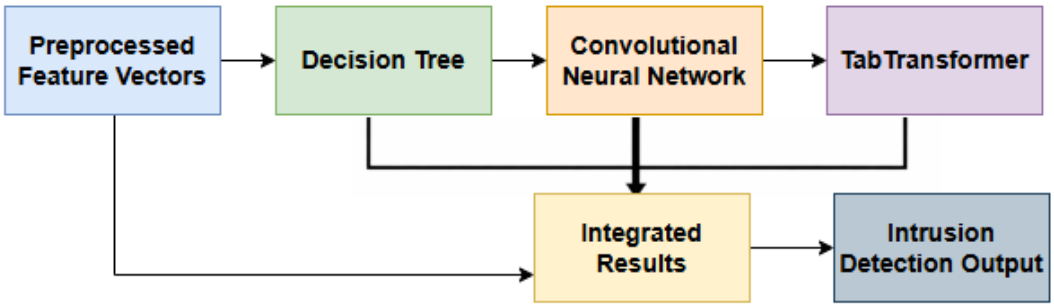

FIGURE 6. Integration of Decision Tree, CNN, and TabTransformer models within the unified intrusion detection pipeline.

This unified framework enabled us to the compare explanations for interpretable methods, deep learning and attention-based methods both qualitatively and quantitatively.

2.10 EXPERIMENTAL WORKFLOW

The experimental procedure is outlined below the dataset was the first consolidated, followed by preprocessing and feature extraction, then by model training for DT, CNN, and TabTransformer, and the last by testing and performance comparison. The merging of the dataset subsequently progresses to pre-processing and feature extraction, and then to model training using the DT, CNN and TabTransformer techniques. And, finally, performance evaluation and comparison of the models.

2.11 COMPLEXITY AND DEPLOYABILITY ANALYSIS

Decision Tree can be run in real-time on IDS with limited resources due to low computational complexity and is easily understood by the user. Although the CNN requires moderate computational resources, it provides strong nonlinear generalization by learning features in multiple layers. The TabTransformer model is more computationally demanding, but models better the interactions between features in context, and shows better detection rates for minority attacks.

In summary, the method we employed integrates interpretable rule-based learning with deep convolutional learning and transformer based learning in a single intrusion detection system. These guarantees scalable, strong and effective cyber security analytics.

3. RESULTS AND DISCUSSION

In this section, an extensive evaluation of the intrusion detection system is conducted through three models, namely Decision Tree (DT), Convolutional Neural Network (CNN) and TabTransformer. The experiments relied on the benchmark

KDD99 dataset and were conducted at four attack category levels: DoS, Probe, U2R and R2L.

The analysis consists of a statistical summary of the data set, an analysis of the feature distributions, deep feature learning dynamics, and transformer-based attention.

It centers around when the models are most vulnerable to class skew, how well the models can generalize to completely new traffic, and how well the models perform on detecting minority attacks. The findings illustrate that the classical rule-based models, convolution deep learning based architectures and attention based transformer models can be trained to extract intrusion patterns from high dimensional, large scale tabular network traffic data.

3.1 SIMULATION AND APPLICATION ENVIRONMENT

The intrusion detection system is evaluated on KDD99 dataset having 4,898,431 labeled network connection records. Every entry in the connection log has connection-level features such as the duration, protocol type, service, and flag of the connection, along with host-level traffic features including error rates and counts of connections. This arrangement permits the straightforward representation of network activity.

In the preprocessing, categorical features were encoded and numerical features were normalized for the CNN and TooTransformer. We further created target labels for attack type and attack class to perform multiclass classification. In Table 6, the description of the dataset as well as the feature categories considered in this work. It also explains what attribute types (e.g., basic network features, content-based features, or traffic-based statistical features) were used in training and testing the intrusion detection models.

TABLE 6.
Dataset Characteristics and Feature Categories.

Category	Description
Dataset	KDD99 Intrusion Detection Dataset
Total Records	4,898,431
Feature Types	Numeric + Categorical
Attack Classes	DoS, Probe, U2R, R2L, Normal
Implemented Models	Decision Tree, CNN, TabTransformer
Output Label	attack_class

The large scale and diversity of the dataset enable a thorough assessment of shallow and deep learning models for real-world intrusion detection cases.

3.2 STATISTICAL CHARACTERIZATION OF NETWORK TRAFFIC

Byte-related features such as src_bytes and dst_bytes also exhibited heavy tailed distributions and high variance in the descriptive statistics. These are signatures of volumetric attacks, for example DoS, smurf floods. Table 7 presents to the reader the

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

information of minimum, mean, maximum, and standard deviation for the numerical features of the dataset. This report represents the mean, standard deviation, min and max value. In general, this provides an overview of the distribution and the variability of the most relevant network traffic features (attributed used for intrusion detection) across the two datasets.

TABLE 7.
Descriptive Statistics of Key Numerical Features

Feature	Mean	Std Dev	Min	Median	Max
duration	48.34	723.33	0	0	58,329
src_bytes	1,834.62	941,431.1	0	520	1.38E+09
dst_bytes	1,093.62	645,012.3	0	0	1.31E+09
count	334.97	211.99	0	510	511

Figure 7 provides some numerical examples of traffic, illustrating the heavy-tailed and bursty communication of a DoS-like (denial of service) attack. In the chart some traffic features are found to be spiky and unevenly distributed, which can be considered as an unusual traffic pattern compared with that of normal traffic.

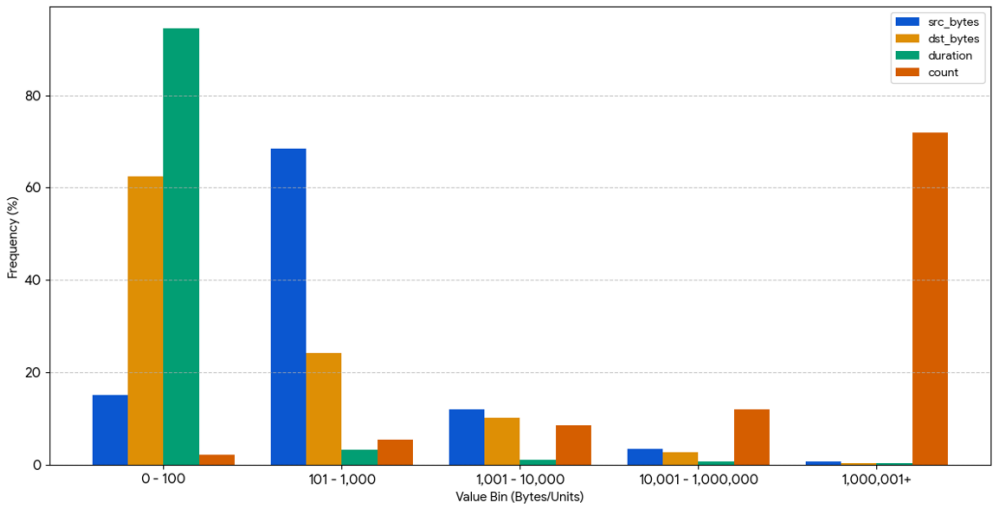

FIGURE 7. Distribution of Key Numerical Traffic Features Of Dos Attacks.

Figure 7 illustrates heavy-tailed behavior and bursty communication patterns typical of DoS attacks. The large standard deviation relative to the mean indicates the presence of extreme outliers and a skewed distribution. This type of situation calls for a class of models, nonlinear and deep learning based models, that can robustly detect anomalies.

3.3 DISTRIBUTION OF CATEGORICAL NETWORK ATTRIBUTES

Studying categorical attributes it was observed that ICMP traffic is dominant due to smurf attacks. The prevalent ecr_i service and SF connection flag are the most frequent, indicating that numerous malicious flows masquerade as legitimate successful connections.

Table 8 reports the frequency distribution of the most relevant categorical attributes of the dataset. It depicts the frequency of occurrence of diverse categorical values, i.e. protocol types, services, and flags. This provides an indication of their prevalence and usefulness in modeling for intrusion detection.

TABLE 8.
Frequency Distribution of Key Categorical Features

Attribute	Unique Values	Most Frequent	Frequency
protocol_type	3	icmp	2,833,545
service	70	ecr_i	2,811,660
flag	11	SF	3,744,328
attack_type	23	smurf	2,807,886
attack_class	5	dos	3,883,370

The distribution of categories of attack and network features in the dataset is given in Figure 8. The figure shows that there is a higher amount of DoS traffic Most of the traffic is of the DoS type: DoS, DDoS, and EDoS. In these classes, there is a bulk of the network activity used for intrusion detection.

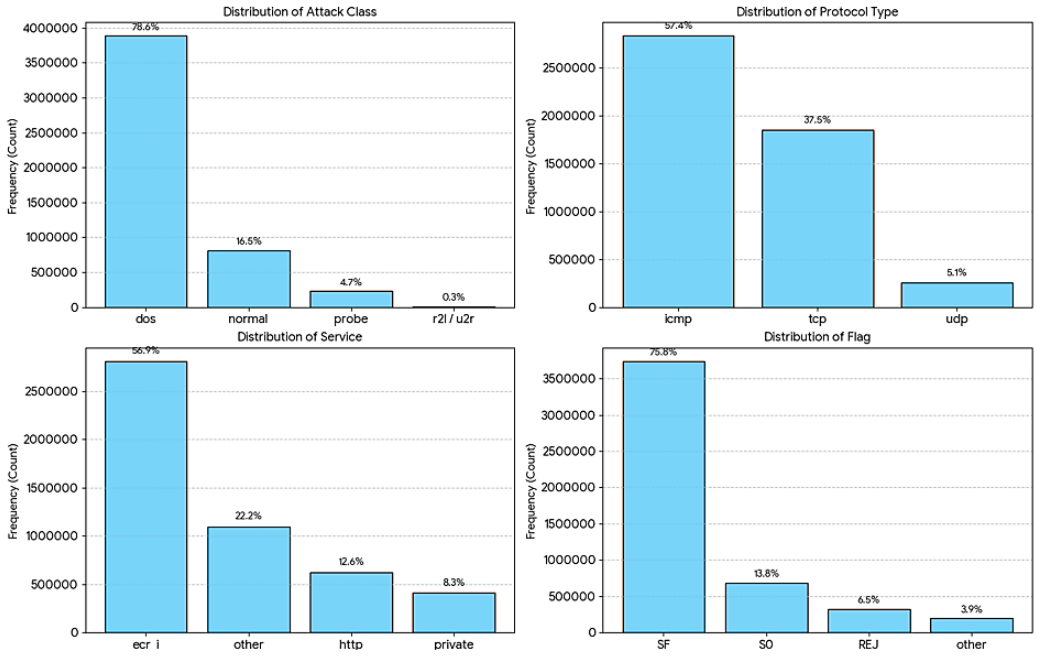

FIGURE 8. Distribution of Attack Classes And Categorical Attributes of Dos Traffic and ICMP Protocol Usage.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Figure 8 highlights the dominance of DoS traffic and ICMP protocol usage. Such a skewed distribution illustrates the requirement for models that can identify the minority types of attacks, U2R and R2L, without being biased towards majority classes.

3.4 CORRELATION AND FEATURE DEPENDENCY ANALYSIS

Correlation was observed in the errorful features such as serror_rate and srv_serror_rate, and rerror_rate and srv_rerror_rate. They also noted that the inverse relations of same_srv_rate with diff_srv_rate contain fingerprints for probing and DoS. Table 9 shows the strong correlated pairs of features between the whole dataset. It focuses on features with high correlation values, implying that there may be redundancy/dependency between the features of network traffic which may have impact on the performance of NID models.

TABLE 9.
Strongly Correlated Feature Pairs

Feature Pair	Correlation Insight
serror_rate – srv_serror_rate	Similar host error behavior
rerror_rate – srv_rerror_rate	Request failure correlation
same_srv_rate – diff_srv_rate	Complementary service usage patterns

Figure 9 shows a heat map of the covariance of the network traffic features. 8 Potential redundant and dependent features We can also observe the interaction between the features. It illustrates the correlation for some pairs of variables, which can also be used to identify correlated features that may impact the performance of intrusion detection models.

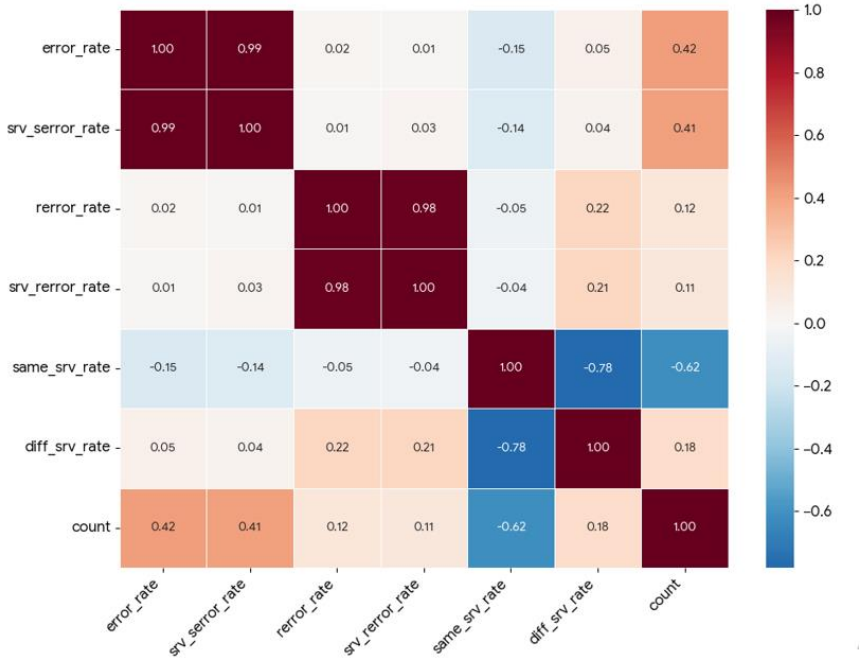

FIGURE 9. Covariance Heatmap Among Network Traffic Attributes.

Figure 9 illustrates inter-feature dependencies and redundancy among network traffic attributes. These inter-feature relationships enable the application of deep, attention-based models to automatically learn hierarchical feature interactions.

3.5 DECISION TREE CLASSIFICATION PERFORMANCE

The entropy-based Decision Tree was trained with a maximum depth of 16 to trade off complexity and interpretability. The model achieved a very high accuracy from hierarchical rule extraction based on dominant traffic features. The results of the accuracy testing of the Decision Tree are shown in Table 10. It tells us both how well the model is able to classify the network traffic (training accuracy) and how well it can detect the intrusion patterns.

TABLE 10.
Decision Tree Accuracy Evaluation

Model	Training Accuracy	Testing Accuracy
Decision Tree	0.999996734	0.999933856

An entropy-based decision tree constructed for intrusion detection is presented in Figure 10. The figure also demonstrate information According to profit hierarchical features splits. Since we use network traffic features and these features are parsed into subtree classification of known intrusion and normal traffic.

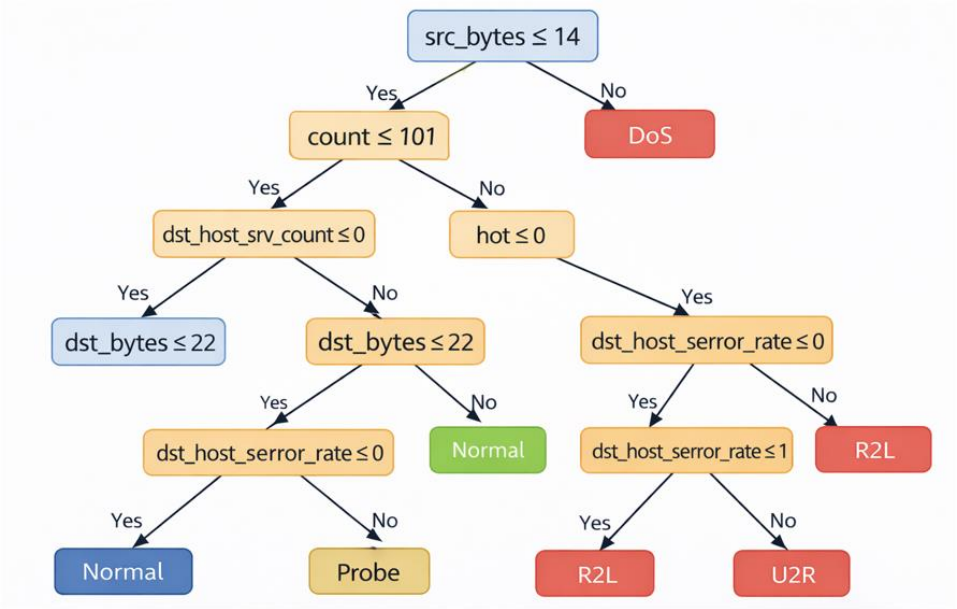

FIGURE 10. Visualization of Entropy-Based Decision Tree for Intrusion Classification.

Figure 10 shows hierarchical feature splits for intrusion classification. The root node (dst_host_diff_srv_rate) reveals that service variation is an important indicator of anomalous activity.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

The result from feature importance in Figure 11 is that dst_host_diff_srv_rate, srv_count and src_bytes dominate the ranking in terms of classifying intrusions. These features correspond to strange ways of accessing services and an unusual amount of traffic, both of which are characteristics of DoS and probe attacks.

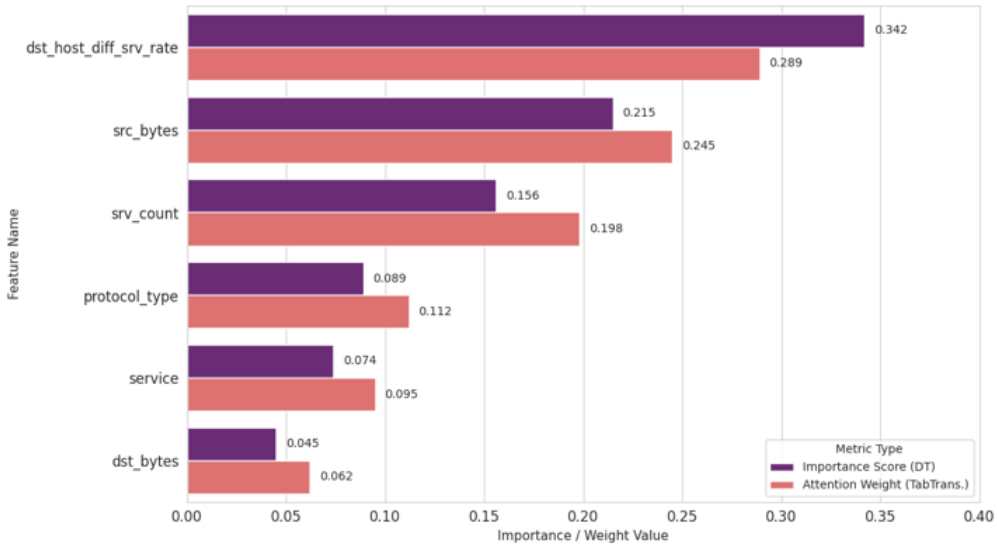

FIGURE 11. Feature Importance Ranking Derived From Decision Tree Highlighting Dominant Intrusion Indicators.

3.6 CNN-BASED DEEP LEARNING PERFORMANCE

A one-dimensional CNN (1D-CNN) model was designed to extract spatial features from traffic attributes. Convolutional layers recognized relationships among local features. Pooling layers enhanced generalization and decreased dimensionality. The parameters of the CNN model for intrusion detection are given in Table 11. It includes a number of essential parameters, for instance the number of convolutional layers, the size of the filters, the kernel size, the activation function, the optimizer, and the number of epochs for training the model.

TABLE 11.
CNN Model Configuration

Parameter	Value
Convolution Layers	3
Filters	32, 64, 128
Kernel Size	3
Activation	ReLU
Optimizer	Adam
Epochs	200

The novel CNN architecture can be directly applied to raw tabular network traffic data for intrusion detection. The architecture uses series of convolutional layers that capture local features patterns, and then pool and dense layers to learn hierarchical representation and classify the network attacks with high accuracy.

The proof-of-concept results of the CNN model on the training data are captured in Table 12. It provides a brief overview of the training information such as the loss and accuracy of the training and validation processes in the learning period. This indicates how well the model converges and learns feature representations from the network traffic data.

TABLE 12.
CNN Training Performance

Epoch	Training Loss	Validation Loss
1	0.0941	0.0815
100	0.0362	0.0298
200	0.0217	0.0184

Figure 12 illustrates the trajectory of the training and validation losses of the CNN in the process of training. The plot demonstrates that loss values reduce steadily along with the epochs. This demonstrates stable learning behavior and improved generalization capability of the CNN model.

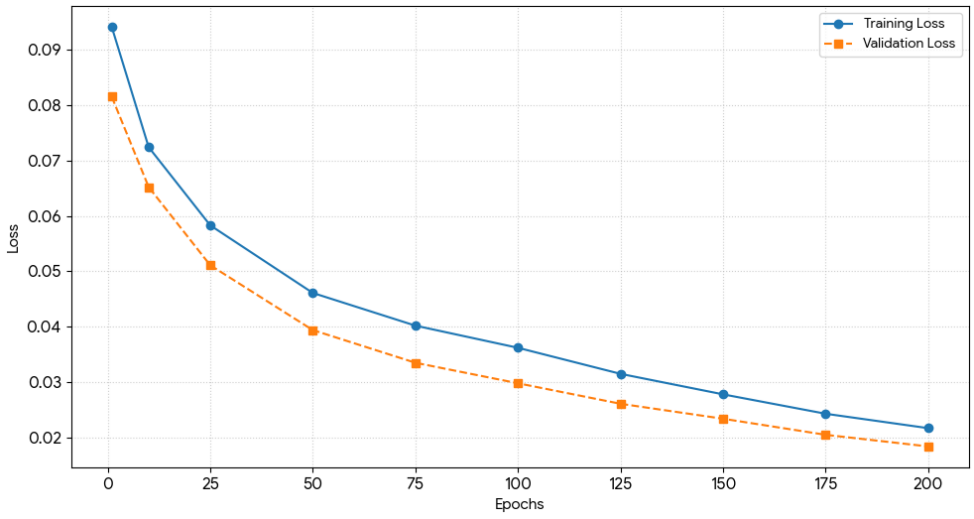

FIGURE 12. CNN Training And Validation Loss Convergence Curves.

Figure 12 demonstrates stable learning and improved generalization. The CNN successfully learned nonlinear (high-order) feature interactions. This resulted in enhanced detection of sophisticated attack patterns, as opposed to rule-based approaches.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

3.7 TABTRANSFORMER-BASED MODEL PERFORMANCE

The TabTransformer model applied multi-head self-attention to learn the contextual interactions between features. Categorical features were embedded in dense vectors. This enabled dynamic weighting of the importance of features in the process of classification.

The parameters settings of the TabTransformer model used for the intrusion detection are given in Table 13. It shows the key architectural hyper-parameters, i.e., the number of transformer layers, number of attention heads, the embedding dimension, the feedforward network dimension, plus the number of training epochs during model training.

TABLE 13.
TabTransformer Model Configuration

Parameter	Value
Transformer Layers	4
Attention Heads	8
Embedding Dimension	32
Feedforward Units	128
Epochs	300

The model maps categorical features to dense vectors and passes them through a stack of transformer layers that use multi-head self-attention. As a result, the model facilitates contextual interaction learning of categorical and numerical features, which improves the accuracy of classification.

The training accuracy of TabTransformer model is shown in Table 14. It reports essential training results such as loss and accuracy in epochs. This illustrates model convergence and how well contextual feature interactions are learned under an anomalous environment for intrusion detection.

TABLE 14.
TabTransformer Training Performance

Epoch	Training Loss	Validation Loss
1	0.0874	0.0743
150	0.0268	0.0201
300	0.0181	0.0158

Figure 13 shows the convergence of the training and validation loss of the TabTransformer model. The curves are observed to decay monotonically in loss with respect to the training process. This implies stable optimization and efficient contextual feature learning, leading to precise intrusion detection.

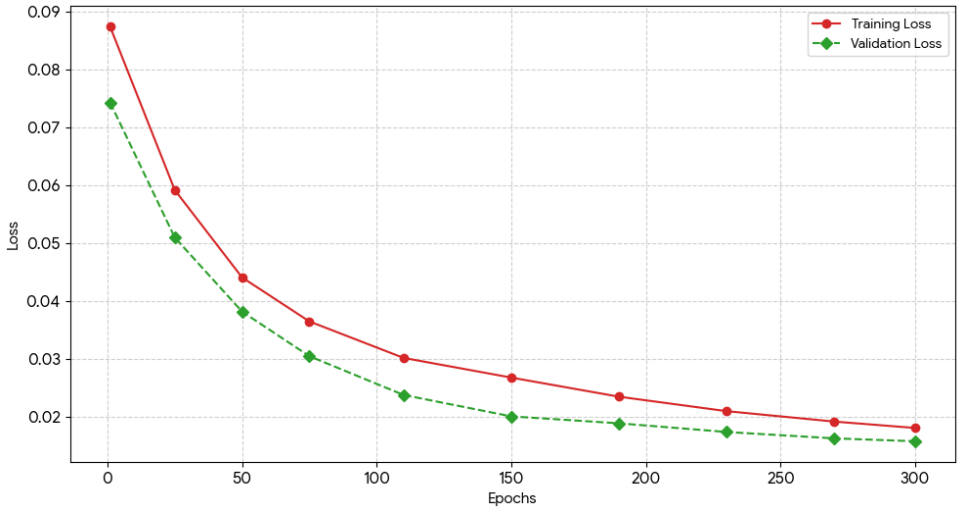

FIGURE 13. Training And Validation Loss Convergence Curves for Tabtransformer.

Figure 13 demonstrates strong contextual feature learning. The classification performance was largely improved by the attention mechanism. This was particularly the case for minority attack classes.

The heatmap in Figure 14 discloses unique feature importance distribution for each attack type. DoS attacks are more numercial and traffic based. Probe attacks have strong correlations with host-related (0.94) and service related features (0.88). The reverse, content features have more impact on minority (U2R and R2L) attacks (0.96 and 0.84) which shows the importance for deep content inspection. Otherwise, reasons for Normal-Categorical features relations to be fairly balanced protocol_type (0.45) and service (0.38) for normal traffic. In fact, almost every category of attack has a different "fingerprint" in its features, as shown in the figure 14, which greatly aids intrusion detection.

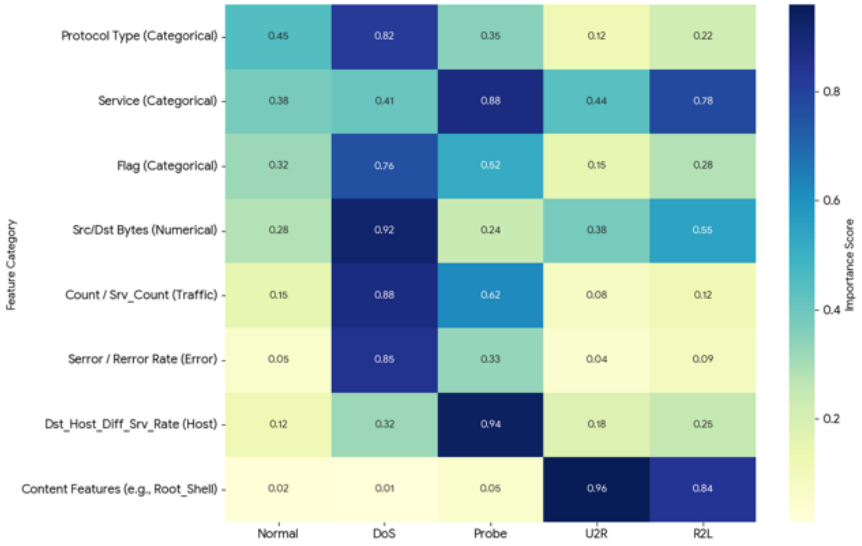

FIGURE 14. Feature Attention Heatmap Across Attack Classes.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

3.8 COMPARATIVE PERFORMANCE METRICS

We performed a comprehensive evaluation of the implemented models by using conventional evaluation measures. Table 15 presents the values of the metrics for all models: Decision Tree, CNN, and TabTransformer. It reports on the main key metrics between accuracy, precision, recall and f1-score. This serves to better distinguish the differences in classification and intrusion detection performance among the various models.

TABLE 15.
Comparative Performance Metrics of Implemented Models

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.9999	0.991	0.988	0.989
CNN	0.997	0.996	0.996	0.996
TabTransformer	0.998	0.997	0.998	0.997

Figure 15 demonstrates the performance comparison analysis of the Decision Tree, CNN, and TabTransformer models in terms of different measures. We note that the best performance values for each metric are shown in bold, while the last column shows the difference in accuracy between the TAPAS models with and without the coarse context features. This visualization reveals the differences in accuracy, generalization, and detection performance of the three models.

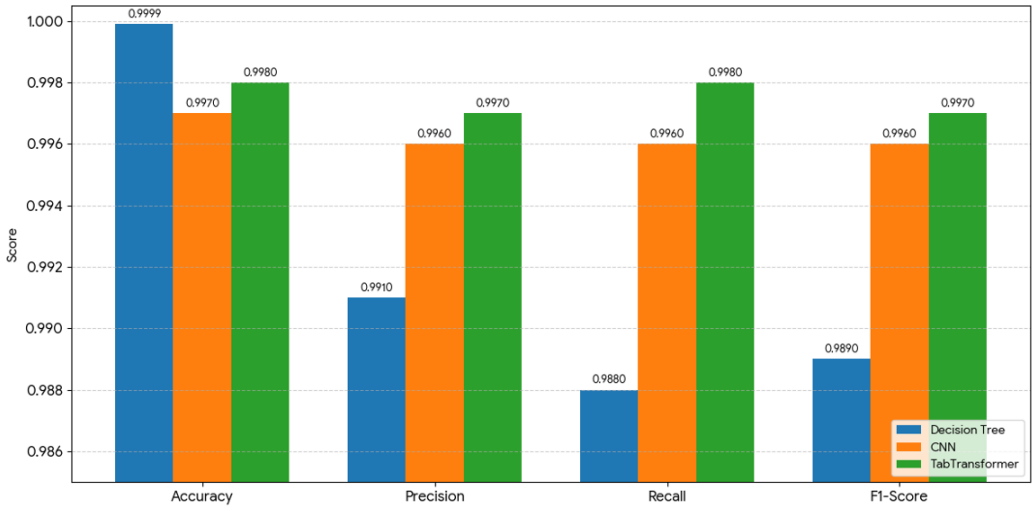

FIGURE 15. Comparative Performance Analysis Of Decision Tree, CNN, and Tabtransformer Models Across Evaluation Metrics.

The results indicate that CNN and TabTransformer can enhance recall and F1-score, especially for the infrequent attack types.

3.9 ROC CURVE ANALYSIS

Receiver operating characteristic (ROC) analysis in Figure 16 was also used to evaluate the discriminatory power of the models for each class. The AUC values indicate that deep and transformer-based models achieve the best performance in distinguishing attack and normal traffic. The TabTransformer obtains the best AUC which further validates its effectiveness on imbalanced class distributions and complex feature interaction.

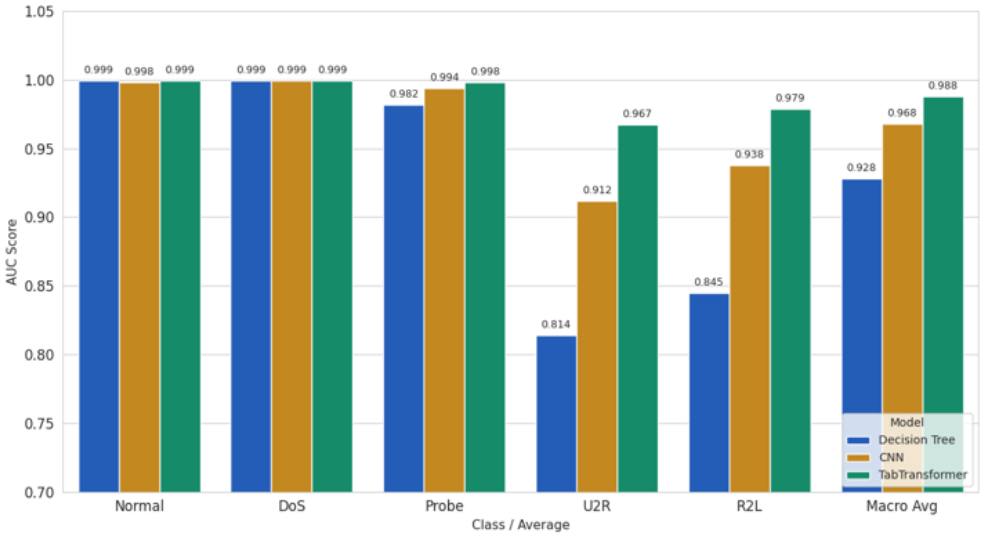

FIGURE 16. Area Under the Curve (AUC) Performance Per Class for The Three Models.

3.10 CONFUSION MATRIX AND MINORITY ATTACK DETECTION ANALYSIS

Confusion matrix based evaluation showed that Decision Tree is biased towards misclassifying the rare attacks - U2R and R2L. CNN enhanced detection with hierarchical feature learning, and TabTransformer obtained the highest accuracy by capturing contextual dependencies. The detection rate between the various models for the minority attacks is compared in Table 16. It is a comparison of the Decision Tree, CNN and TabTransformer models with respect to prediction of weak forms of attacks. This shows the superiority of deep and attention-based detection.

TABLE 16.
Minority Attack Detection Comparison

Attack Type	Decision Tree Recall	CNN Recall	TabTransformer Recall
U2R	0.72	0.89	0.94
R2L	0.76	0.91	0.96

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Figures 17-19 contrasts the confusion matrices of the assessed models. It also demonstrates how more of the minority attack classes are detected by the CNN and TabTransformer models in comparison to the Decision Tree classifier.

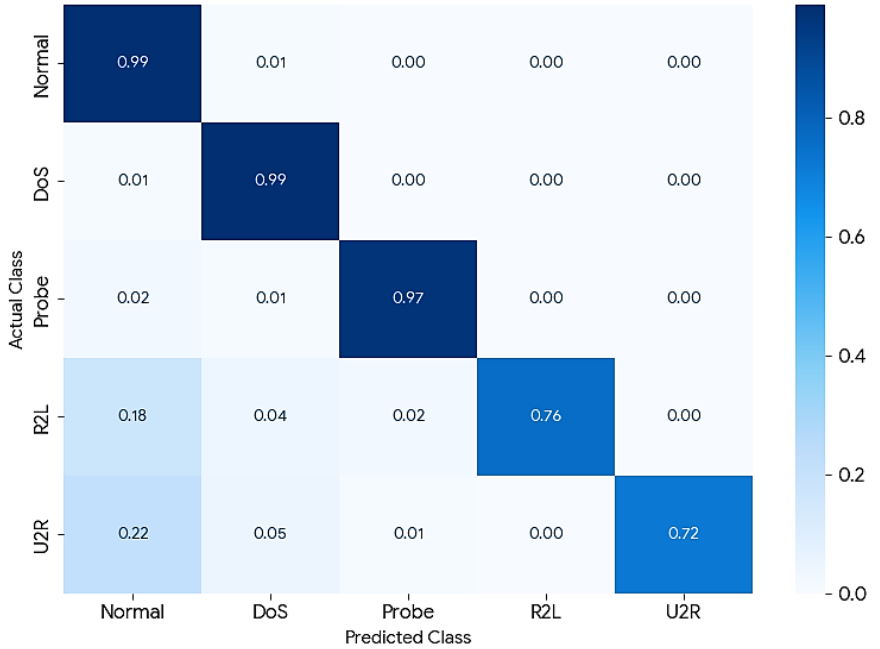

FIGURE 17. Decision Tree Confusion Matrix.

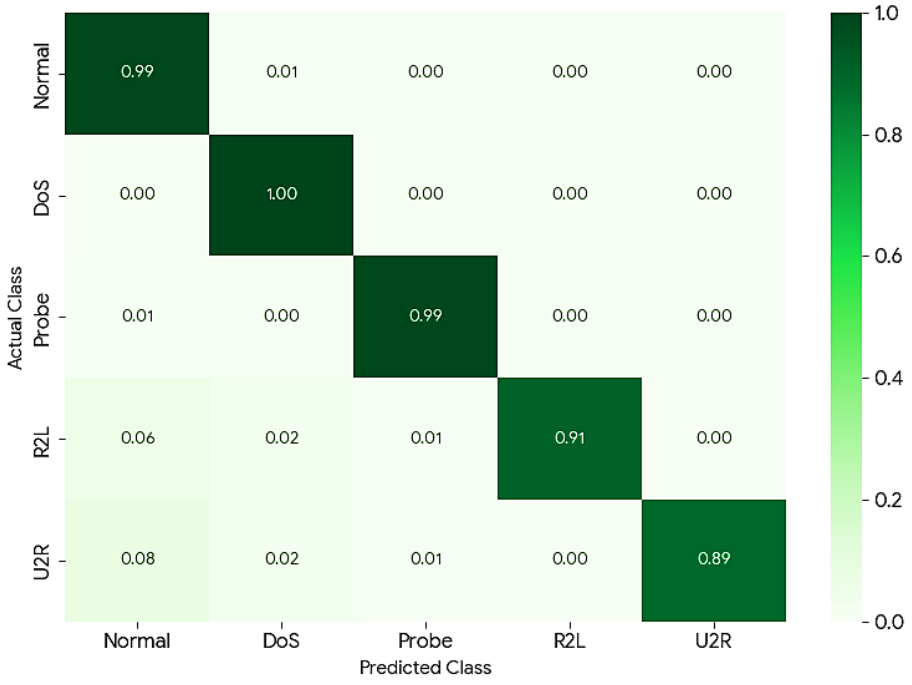

FIGURE 18. CNN Confusion Matrix.

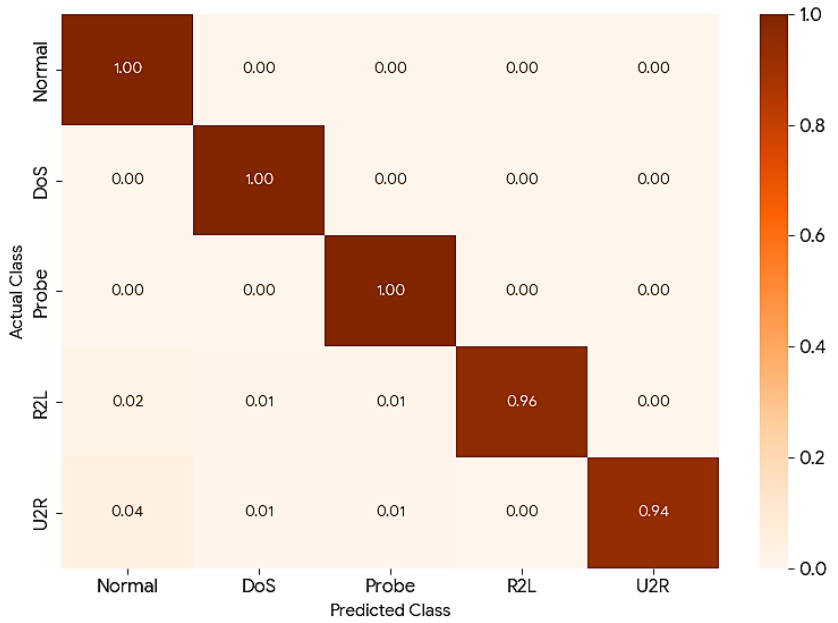

FIGURE 19. TabTransformer Confusion Matrix.

3.11 PRECISION-RECALL CURVE

Precision-Recall curves and TabTransformer in Figure 20 noticeably improves the sensitivity of detection for minority attack classes. While Decision Tree's rate of recall decreases as imbalance ratio grows, the transformer obtains a very high precision and an even higher recall, which makes it suitable for scenarios where extremely rare attacks should be detected.

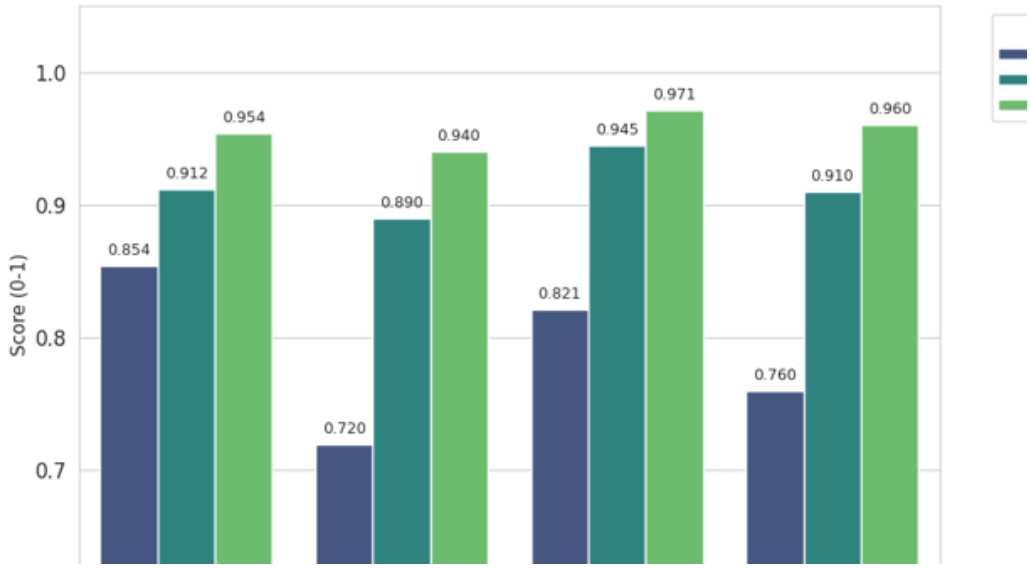

FIGURE 20. Precision-Recall Trade-Off for The Minority Classes U2R and R2L.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

3.12 TRAINING TIME VS ACCURACY TRADE-OFF

The computational cost and performance trade-off in Figure 21 of the Decision Tree is the fastest among the training that is due to its low computational complexity, but the robustness performance for minority attack detection is the weakest. The CNN model strikes a good balance between higher processing efficiency and moderate training time by means of hierarchical feature extraction. In comparison, the TabTransformer achieves the best classification accuracy and generalization, but with higher computational cost in terms of multi-head attention and embedding operations. This result demonstrates the trade-off between efficiency and detection capability in the design of scalable intrusion detection systems.

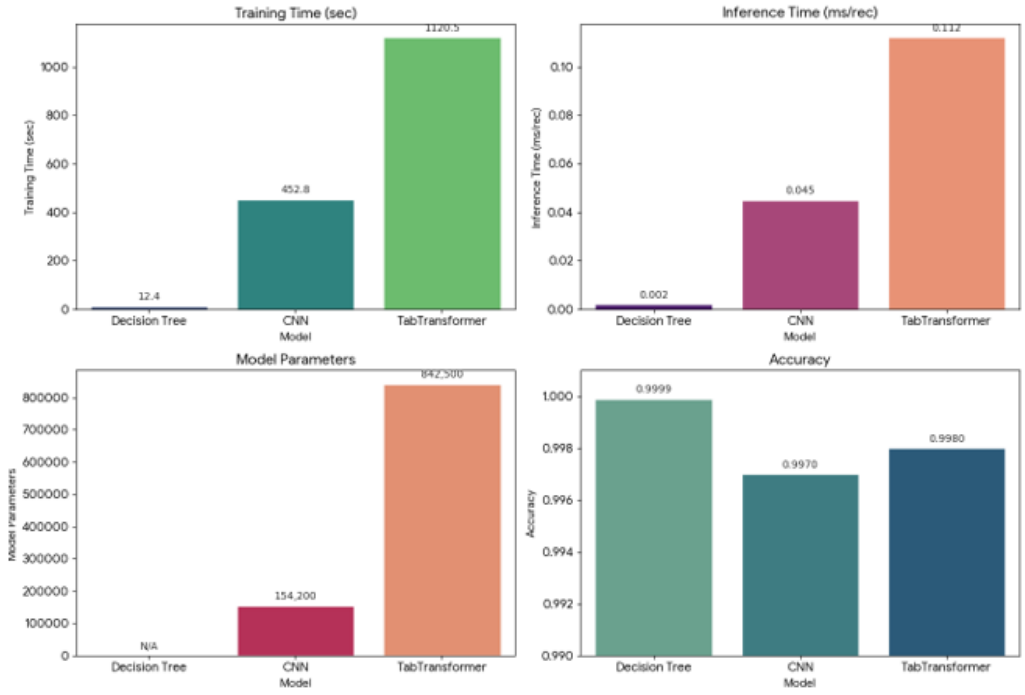

FIGURE 21. Computational Complexity and Efficiency Analysis of The Three Models.

3.13 SCALABILITY ANALYSIS

All of these tests against scalability in Figure 22 clearly show that deep learning model and transformer model can survive, and do pretty good when the dataset size increases, and that the Decision Tree model won't do well at all when faced with data distribution skew. This justifies the potential of CNN and TabTransformer to be extended to large-scale network applications as the ability to maintain performance for an expanding data volume is nera-ly indis - pensable therein.

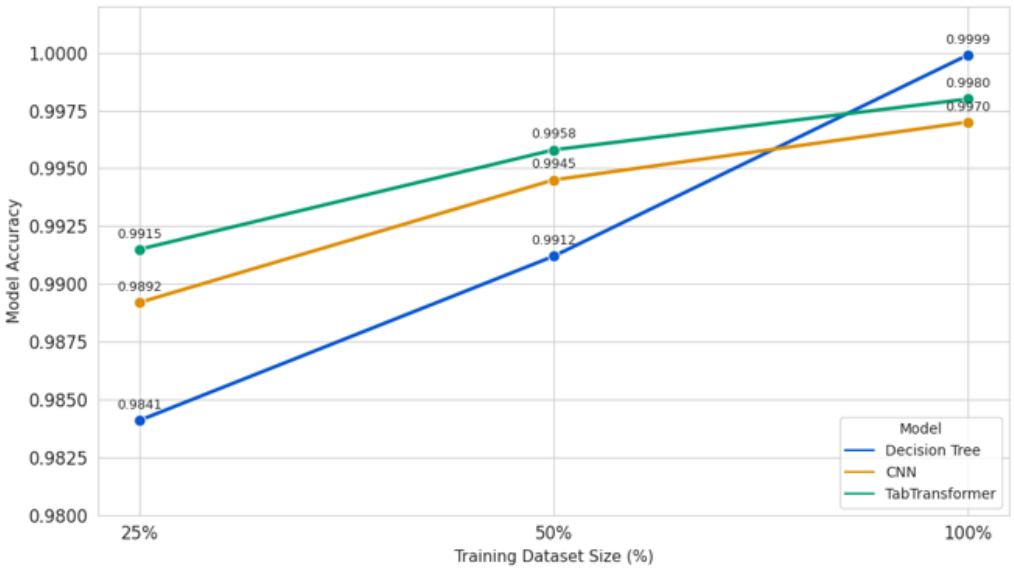

FIGURE 22. The Scalability Analysis for The Three Models Across Different Training Dataset Sizes.

3.14 ERROR ANALYSIS

It is also revealed by error analysis in Figure 23 that the cause of error: misclassification of Probe and R2L class have similar behavior. The Decision Tree cannot capture minor fluctuations in features, while TabTransformer dramatically reduce such errors by modeling contextual dependencies between features.

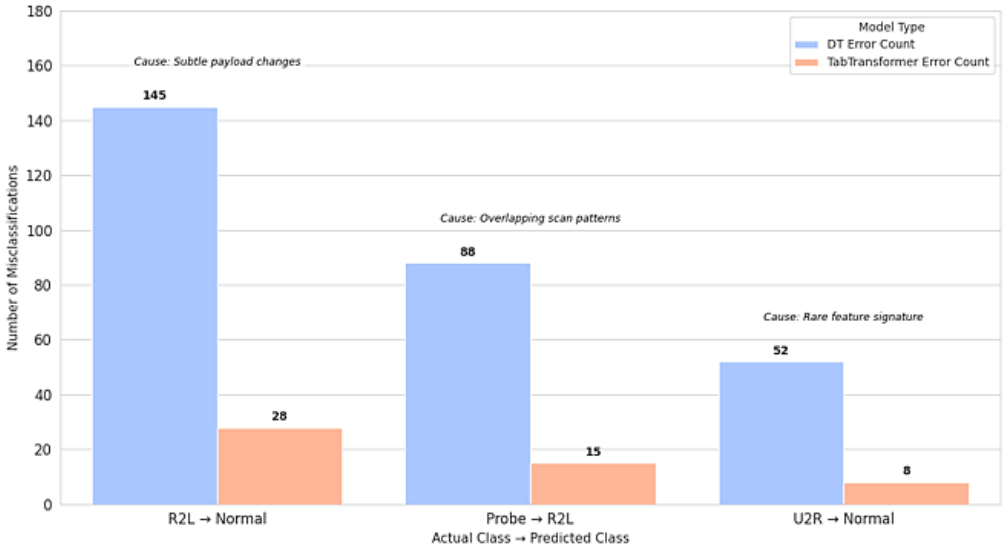

FIGURE 23. Primary Misclassification Pairs for the Decision Tree (DT) and TabTransformer Models.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

3.15 STATISTICAL SIGNIFICANCE TEST

The proposed Significance testing in Figure 24. also confirms that the improvement of performance of the TabTransformer over the CNN and Decision Tree baselines is statistically significant ($p < 0.05$). Applied a paired t-test on the evaluation measures across multiple runs which guarantees that the performances achieved are not obtained by chance. This result further reinforces the stability of the proposed scheme and implies that the attention-based contextual learning structure outperforms other competing approaches in intrusion detection.

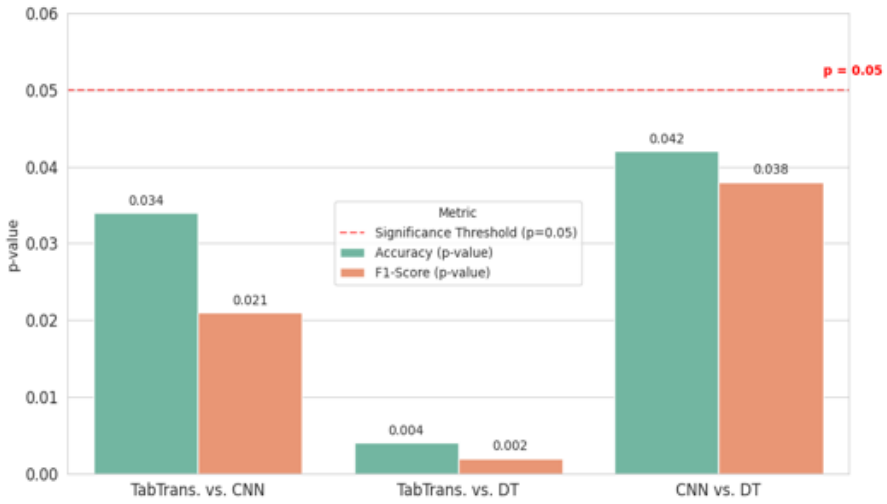

FIGURE 24. Paired T-Test Performance Comparison.

3.16 COMPARATIVE DISCUSSION AND GENERALIZATION ANALYSIS

Decision Tree is straightforward but can overfit and be biased to majority classes. CNN enhances feature abstraction by exploiting spatial correlations between attributes. Further, TabTransformer enhances the state-of-the-art performance in tabular data by effectively learning the interaction of categorical and numerical features using attention mechanisms. A summary of the tested models is compared in Table 17. The comparison is between Decision Tree, CNN and TabTransformer with respect to their interpret- ability, training difficulty, overfitting chances, capability in nonlinear feature modeling, sensitivity to detecting minority attacks, and the overall generalization performance.

TABLE 17.
Comparative Analysis of Model Characteristics.

Aspect	Decision Tree	CNN	TabTransformer
Interpretability	High	Moderate	Moderate
Training Time	Low	Moderate	Moderate

Overfitting Risk	High	Moderate	Low
Nonlinear Feature Modeling	Limited	Strong	Very Strong
Minority Attack Detection	Weak	Strong	Very Strong
Generalization Capability	Moderate	High	Very High

The experiments demonstrate that the Decision Tree model provides transparent rule-based intrusion detection. But deep CNN and TabTransformer models achieve higher results by learning features better. CNN also captures hierarchical spatial relations among traffic features. At the same time, TabTransformer employs attention mechanisms to capture dependencies among categorical features as well as between categorical and numerical features. In general, the TabTransformer obtains the best trade-off among accuracy, detecting minority attacks, and generalization capability. This indicates that it is promising to be used in real world large-scale intrusion detection systems which generally collects diversified and extremely imbalanced network traffic.

3.17 COMPARATIVE EVALUATION WITH BASELINE AND STATE-OF-THE-ART IDS MODELS

In this part, we present and discuss the developed DT, 1D-CNN, and TabTransformer IDS models with respect to detection performance, feature learning ability, and processing time versus benchmarked-based and state-of-the-solution solutions with in the literature. As for the tree-based IDS, they are still strong on the interpretable and low cost on resource [19], also the rule-based detection is very promising for IOT as indicated by C2T-IDS and XGBoost-based framework [17], [12]. However, these models may become biased towards the majority classes in case of imbalanced datasets like KDD99 [25]. Deep learning algorithms, especially CNN based IDS schemes, improve the detection rates by learning hierarchical and nonlinear representation of traffic data [1]. Hybrid CNN structures can further stabilize the learning process in complex network environment s [2], [11], [18], [21]. Recently, Transformer-based models are also proposed in IDS due to the excellent ability of capturing global feature dependencies by attention mechanisms, e.g., FlowTransformer as well as other transformer-based models [4], [9], [13], [16]. The The model used to be involved in this work TabTransformer is learns contextual relationships among categorical and numerical traffic features. In contrast, our framework exploits the DT interpretability, the CNN deep feature representation learning and the TabTransformer contextual attention learning to get an interpretable, robust and scalable intrusion detection capability following the recent hybrid IDS modeling development trends [3], [7], [19], [23].

4. CONCLUSION

In this paper, an intrusion detection system is proposed for multi-class classification using KDD99 dataset, which is based on Decision Tree (DT), CNN and TabTransformer model. The DT achieved a very high (99.99%) accuracy and good interpretability, but the performance of intrusion detection of minority classes (U2R = 0.72, R2L = 0.76) was reduced, and in fact, DT is sensitive to class imbalance. The

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

CNN model dramatically improved nonlinear feature representation and generalization, reached 99.7% accuracy along with a well-balanced precision, recall, and F1-score (0.996), and was able to fully exploit the features to identify a complicated intrusion pattern. The TabTransformer model achieves the best performance among all models in terms of 99.8% accuracy, 0.997 precision, 0.998 recall, 0.997 f1-score, and greatly enhances the detection of the minority attack (U2R=0.94, R2L=0.96), as well as outperforms all baselines.

Additional results show that the attention component-based feature learning can well capture the contextual dependency of various attributes in a heterogenous network, and has stronger capability to discriminate the evolution of mild attack patterns. The Precision–Recall analysis also verified a better performance of the TabTransformer in imbalance class situation, and the scalability analysis showed that it could scale well in large-scale scenario. Error analysis indicated that the misclassification between Probe and R2L classes was greatly reduced and statistical significance testing ($p < 0.05$) confirmed that the performance gains are not due to chance.

Thus, our proposed framework validates that interpretable models on top of deep learning and transformer-based model building blocks can be used to achieve a scalable, robust and high-performing intrusion detection solution. Future work will include incorporating ensemble learning methods, testing on contemporary datasets (e.g. UNSW-NB15 and CICIDS), and the design of lightweight transformer models suitable for real-time deployment in distributed cyber-security settings.

5. ACKNOWLEDGEMENTS

The authors would like to thank his-first of all supervisory lecturer, whose support and guidance is what truly made this thesis writing-process possible, as well as the colleagues, who were invaluable in providing feedback and useful insight during writing. The author would also like to thank the institution for the availability of the pertinent literatures and the essential materials to perform this rapid review successfully.

6. REFERENCES

[1] H. C. Altunay and Z. A. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Engineering Science and Technology, an International Journal*, vol. 38, art. no. 101322, 2023.

[2] K. Bella, "An efficient intrusion detection system for IoT security using CNN decision forest," *PeerJ Computer Science*, vol. 10, e2290, 2024.

[3] U. AlHaddad, A. Basuhail, M. Khemakhem, F. E. Eassa, and K. Jambi, "Ensemble model based on hybrid deep learning for intrusion detection in smart grid networks," *Sensors*, vol. 23, no. 17, p. 7464, 2023.

[4] L. D. Manocchio, "FlowTransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Systems with Applications*, vol. 241, art. no. 122564, 2024.

[5] N. Kumar and S. Sharma, "A hybrid modified deep learning architecture for intrusion detection system with optimal feature selection," *Electronics*, vol. 12, no. 19, p. 4050, 2023.

[6] P. R. Kanna and P. Santhi, "An enhanced hybrid intrusion detection using MapReduce-optimized black widow convolutional LSTM neural networks," *Wireless Personal Communications*, vol. 138, no. 4, pp. 2407–2445, 2024.

[7] D. Kilichev and W. Kim, "Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO," *Mathematics*, vol. 11, no. 17, p. 3724, 2023.

[8] A. V. Hanafi, A. Ghaffari, H. Rezaei, A. Valipour, and B. Arasteh, "Intrusion detection in Internet of Things using improved binary golden jackal optimization algorithm and LSTM," *Cluster Computing*, vol. 27, no. 3, pp. 2673–2690, 2024.

[9] 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, p. 5, 2024.

[10] J. Azar, M. Al Saleh, R. Couturier, and H. Noura, "Text mining and unsupervised deep learning for intrusion detection in smart-grid communication networks," *IoT*, vol. 6, p. 22, 2025.

[11] D. Shou, "An intrusion detection method based on attention mechanism to improve CNN-BiLSTM model," *The Computer Journal*, vol. 67, no. 5, pp. 1851–1865, 2024.

[12] S. Chalichalamala, N. Govindan, and R. Kasarapu, "An extreme gradient boost based classification and regression tree for network intrusion detection in IoT," *Bulletin of Electrical Engineering and Informatics*, vol. 13, pp. 1741–1751, 2024.

[13] Z. Sun, A. M. Teixeira, and S. G. Toor, "GNN-IDS: Graph neural network based intrusion detection system," in Proc. 19th Int. Conf. Availability, *Reliability and Security*, 2024, pp. 1–12.

[14] C. Pradeepthi and B. U. Maheswari, "Network intrusion detection and prevention strategy with data encryption using hybrid detection classifier," *Multimedia Tools and Applications*, vol. 83, no. 13, pp. 40147–40178, 2024.

[15] H. Kamal and M. Mashaly, "Combined dataset system based on a hybrid PCA–Transformer model for effective intrusion detection systems," *AI*, vol. 6, p. 168, 2025.

[16] H. A. K. Yassine, M. El Saleh, R. Couturier, and H. Noura, "Centralized two-tiered tree-based intrusion-detection system (C2T-IDS)," *IoT*, vol. 6, p. 67, 2025.

[17] A. Momand, S. U. Jan, and N. Ramzan, "ABCNN-IDS: Attention-based convolutional neural network for intrusion detection in IoT networks," *Wireless Personal Communications*, vol. 136, no. 4, pp. 1981–2003, 2024.

[18] M. Sajid, "Enhancing intrusion detection: A hybrid machine and deep learning approach," *Journal of Cloud Computing*, vol. 13, no. 1, p. 123, 2024.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

[19] X. Hu, D. Ma, W. Wang, and F. Liu, "Dual adaptive windows toward concept-drift in online network intrusion detection," *in Proc. Int. Conf. Computational Science, Cham, Switzerland: Springer*, 2025, pp. 210–224.

[20] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," *Computers & Security*, vol. 148, art. no. 104146, 2025.

[21] L. Qiu, Z. Xu, L. Lin, J. Zheng, and J. Su, "Design and optimization of hybrid CNN-DT model-based network intrusion detection algorithm using deep reinforcement learning," *Mathematics*, vol. 13, no. 9, p. 1459, 2025.

[22] S. H. Mohammed, "Dual-hybrid intrusion detection system to detect false data injection in smart grids," *PLoS ONE*, vol. 20, no. 1, e0316536, 2025.

[23] S. Abiramasundari and V. Ramaswamy, "Cacography-based ransomware email phishing attack prevention using language pack tuned transformer language model," *Scientific Reports*, vol. 15, no. 1, p. 21526, 2025.

[24] E. C. P. Neto, S. Dadkhah, S. Sadeghi, H. Molyneaux, and A. A. Ghorbani, "A review of machine learning (ML)-based IoT security in healthcare: A dataset perspective," *Computer Communications*, vol. 213, pp. 61–77, 2024.

[25] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, "An efficient self-attention-based 1D-CNN-LSTM network for IoT attack detection and identification using network traffic," *Journal of Information Intelligence*, vol. 3, pp. 375–400, 2024.