Open Access Original Article

Encrypted Network Traffic Classification Using Intelligent Techniques

Shudhamati Mali [1] , Mansi Gujral [1] , Aswani Kumar Cherukuri [1]

1. Information Technology, Vellore Institute of Technology, Vellore, IND

Corresponding authors: Shudhamati Mali, shudhamatimali15@gmail.com, Aswani Kumar Cherukuri, cherukuri@acm.org

Review began 11/20/2024
Review ended 01/06/2025
Published 01/16/2025

DOI: https://doi.org/10.7759/s44389-024-02701-2

Abstract

Traffic classification is considered one of the central components of network management and security operations, as it covers significant aspects such as traffic prioritization, anomalous behavior identification, and security policy implementation. Methods for traffic classification based on port and payload analysis are becoming less effective due to the nature of contemporary applications: dynamic ports, encrypted traffic, and complex protocols. To address these issues, two promising approaches have gained popularity: the use of machine learning and deep learning methods, which allow the identification of traffic sources based on patterns rather than relying on protocols and ports. In this study, the efficiency of various models is assessed, including support vector machines, random forests, convolutional neural networks (CNNs), recurrent neural networks (RNNs), CNN-RNN, and long short-term memory networks. This work indicates that deep learning models such as CNNs and RNNs have outperformed traditional machine learning models. Among them, the model based on the RNN architecture achieved the highest accuracy of 93% across all datasets. The random forest model also performed strongly, with notable improvements in precision and recall. This paper offers important information that will be useful for improving existing models and developing new models for network traffic classification, which is the basis of intelligent security systems.

Categories: Network Security, Cryptography, Machine Learning (ML)
Keywords: deep learning, intelligent techniques, machine learning, encrypted network traffic classification, performance

Introduction

Balancing the security with efficiency while managing network traffic in today's digital landscape becomes important. With the increasing size and speed of networks, older techniques such as port number checking and packet sniffing are proving to be ineffective because most modern technologies utilize encryption and dynamically allocated ports. Contrasting this view, the intelligent techniques have shown very good promise by classifying network traffic using pattern and statistical characteristics, rather than depending on specific protocols or contents of data packets [1,2]. Thus, these approaches classify both normal network activities and malicious actions such as spam or cyber attacks that usually remain undetected through traditional detection systems. Although various machine learning and deep learning models have been explored in the existing literature for network traffic classification, there is a large research gap in their performance assessment on different datasets, particularly for encrypted traffic. To fill this gap, this study takes an initiative by using a variety of datasets and applying different algorithms, including machine learning models like support vector machines (SVM), random forest, decision tree, and k-nearest neighbors (k-NN), and deep learning models like convolutional neural networks (CNN), recurrent neural networks (RNN), CNN-RNN hybrids, and long short-term memory (LSTM). Through a systematic analysis of their performance, this paper attempts to gain deeper insights into the applicability of intelligent techniques in distinguishing benign traffic from potentially malicious activities such as spam or attacks.

This study compared machine learning models together with deep learning models, such as decision trees, random forest, SVM, k-NN, and neural networks [3]. These experiments are performed on three core datasets: UNSW-NB15, which contains normal and synthetic attack traffic; ISCXTor2016, simulating realistic Tor network traffic; and the phishing dataset for identifying evil websites. Finally, this paper investigated how the data processing approach, like feature extraction and the combination of multiple algorithms, may improve the classification accuracy. This work focuses on the way such large data volumes can be handled since network traffic has to be analyzed in real time.

This paper tries to understand the basis on which intelligent techniques help solve the complexities of modern network traffic. In that regard, the main contributions of this paper are as follows:

1. Highlighting the importance of feature selection in classifying such network traffic.

2. Evaluating the effectiveness of machine learning and deep learning models for traffic classification.

How to cite this article
Mali S, Gujral M, Cherukuri A (January 16, 2025) Encrypted Network Traffic Classification Using Intelligent Techniques. Cureus J Comput Sci 2 : es44389-024-02701-2. DOI https://doi.org/10.7759/s44389-024-02701-2

3. Demonstrating critical results showcasing how intelligent techniques contribute to more robust network environments and improved cybersecurity [4].

The remainder of the paper is structured as follows. Literature review section focuses on related work in the field of network traffic classification and security. Materials and methods section elaborates on the machine learning and deep learning models. Architecture and experimental setup section describes the datasets and workflow. Results section presents the results, their interpretation, and their significance in the context of network security. Finally, Conclusions section concludes the paper by summarizing the key findings, limitations, and proposing directions for future research.

Literature review

A rapidly increasing number of internet-connected devices and the growth of cloud computing have caused a particularly steep rise in both the amount of network traffic and its complexity. This torrential rise in traffic makes network management critical to robust security, optimal resource utilization, and overall performance [5]. Over time, port-based and signature-based techniques have dominated the field for years but fail to address many of the complexities involved. The growing use of encrypted communication protocols adds another layer of complexity and imposes considerable limitations on traditional traffic-analysis techniques. These considerations have stimulated increased interest in applying intelligent techniques to network traffic classification. Deep learning, a family of machine learning methods using multi-layered neural networks, can recognize very complex patterns and abstract features that would otherwise be rejected from large datasets [6]. There is great potential in the development of highly adaptive and accurate traffic classification systems that work in real time, even in the presence of certain encryption or obfuscation techniques. It would develop a deep learning model for network traffic classification and explore methodologies to exploit such models to enhance both accuracy and scalability over the current approaches.

Recent studies have shown the integration of hybrid approaches and optimization techniques for improving network traffic classification. Dakic et al. [7] discuss metaheuristic optimization techniques for intrusion detection in Internet of Things (IoT) and Industrial Internet of Things (IIoT) environments, with a particular focus on software in autonomous vehicles. The research points out the specific security challenges posed by these systems due to their high connectivity and limited computational resources. The study uses metaheuristic algorithms for feature selection and optimization to achieve enhanced detection accuracy and efficiency. With the proposed system, adaptation to the dynamic nature of IoT/IIoT traffic has been demonstrated, making the contribution to securing these domains highly critical. Khafaga et al. [8] introduce a new voting classifier for network intrusion detection that integrates an ensemble of optimized machine learning models with a modified whale optimization algorithm (WOA). The new WOA is guided by the dipper-throated optimizer to enhance exploration during optimization [8]. It assesses the proposed voting classifier using a dataset derived from IoT devices and proves its superiority in classifying robust and efficient intrusion. The metaheuristic optimization improves the performance of the standard machine learning model and significantly impacts real-time network traffic analysis. Savanović et al. [9] address security problems related to Healthcare 4.0 systems using the machine learning models optimized with a modified Firefly Algorithm. The authors demonstrate the potential of metaheuristic optimization to solve NP-hard problems in real-time intrusion detection, which is critical for sustainable healthcare IoT systems. By exploring the factors contributing to security issues using SHapley Additive exPlanations (SHAP), they can improve intrusion patterns and, ultimately, the whole system security.

This paper draws on the inherent ability of deep neural networks to learn complex patterns directly from raw network traffic data, with the aim of overcoming some of the limitations that legacy approaches have recently encountered in traffic analysis [10]. As such, the real aim is to improve the security and performance of networks through classification systems that are robust, scalable, and flexible enough to cope with the complexity of modern digital communications.

The fast-changing digital environment makes network traffic management and security increasingly challenging due to the sheer volume and complexity of the current data traffic [11]. Another issue, which is widely growing today, is encrypted traffic between legitimate and malicious activity, which makes this differentiation much harder. These challenges and the need for creative solutions call for the work. Advanced network traffic classification can be achieved by using intelligent techniques that directly learn from raw data. Making use of deep neural networks, this article aimed to design a highly adaptive traffic classification system in response to the nuances of the present-day network architectures. The key motivational drive behind this work would be to equip the network administrators and security experts with real-time tools that would facilitate an easy way of monitoring and classifying network traffic while also enabling them to trace probable threats in very quick and accurate ways. Finally, the objective of this work would be to bridge the gap that currently exists between traditional traffic classification methods and the requirements of complex network environments and, thus, make a contribution towards more secure, efficient, and resilient infrastructure of networks.

Network traffic classification research has seen enormous growth over the last decade because of the continued need for improving both network security and functionality. This literature review follows a natural progression in the history of how classifications have evolved from simple, traditional methods involving the use of port numbers and packet payloads to the current version relying on machine learning algorithms. Early work, such as that of Moore and Zuev [12], demonstrates the applicability of these techniques with traditional usage of network protocol conventions for port numbers and unencrypted data. However, relatively more recent work, including that of Auld et al. [13], indicates encryption and dynamic allocation of port numbers significantly lessened the utility of those older methods so a new generation of more resilient and sophisticated methods emerged.

These limitations led to the integration of machine learning into traffic classification. Nguyen and Armitage [14] were part of a pioneering team that used elementary machine learning algorithms, including Naive Bayes and k-NN, to classify traffic on the basis of statistical features extracted from flow data. Their pioneering work opened up ways of using machine learning to overcome some of the limitations associated with manual feature selection and protocol-specific rule dependency. Williams et al. [15] compared five algorithms for IP traffic flow classification in terms of their accuracy, speed, and scalability. Their study furnished crucial information regarding the practical applicability of such methods in network traffic analysis.

With increased computational strength, deep learning has now been discovered as a quintessential component in traffic classification. Yang et al. [16] suggested the utilization of CNN for packet sequence analysis that exceeded the performance of several previous techniques that could not identify most of the encrypted patterns. Similarly, AlEroud and Karabatis [17] applied RNN to represent temporal dependencies in the network traffic, further strengthening the classification accuracy across dynamic network environments.

Aouedi et al. [18] proposed a deep learning-based ensemble model that integrates decision tree classifiers using a nonlinear blending approach to enhance network traffic classification. It outperforms traditional machine learning models by showing increased accuracy and generalization across datasets. Xue et al. [19] discuss the growing challenges in traffic classification with an increasing network bandwidth, complicated applications, and sophisticated evasion techniques. Jovanovic et al. [20] proposed a hybrid approach to detect phishing websites that combines the XGBoost model with a two-tier metaheuristic optimization framework. Feature selection and hyperparameter tuning have improved the Firefly algorithm. Hence, it considerably enhances the model's performance. The authors performed the system's evaluation on three publicly available phishing website datasets that show the proposed method outperforms other approaches. The use of SHAP analysis also helps to drill down into the contribution of each feature toward the detection process.

Rachmawati et al. [21] addressed the limitations of traditional approaches to network traffic classification, such as port-based and payload-based methods, especially in terms of processing encrypted traffic. The paper emphasized that deep learning could overcome these challenges by relying on automatic feature learning, improving precision, and bypassing the complexities of modern network traffic.

Martínez Torres et al. [22] reviewed machine learning in cyber security, concentrating on its challenge of solving issues with intrusion and anomaly detection. Zivkovic et al. [23] proposed a new hybrid Firefly algorithm to optimize XGBoost hyperparameters for network intrusion detection, which helps to overcome problems such as high false positives and false negatives. Their approach, tested on NSL-KDD and UNSW-NB15 datasets, outperformed traditional methods and other metaheuristics in classification accuracy and precision. This work shows the potential of metaheuristic optimization to improve the performance of machine learning classifiers in NIDS.

Materials And Methods

Intelligent techniques can be of two types: deep learning and machine learning. In this research, both techniques have been applied on the given dataset.

Machine learning-based models

SVM is a powerful supervised learning algorithm used for classification tasks [24]. In this research, SVM is employed with various kernel functions, including linear, polynomial, sigmoid, and radial basis function. SVM is well suited for binary and multi-class classification problems, making it suitable for network traffic classification tasks. By defining an optimal hyperplane or decision boundary, SVM aims to maximize the margin between different classes, leading to robust classification performance. The choice of different kernel functions allows for the exploration of the non-linear separability of network traffic data and the assessment of the impact of kernel selection on classification accuracy.

Random forest is a powerful ensemble learning method that is particularly suitable for classification problems, such as network traffic classification. It constructs several decision trees and, at runtime, outputs the class that is the majority vote of the individual trees, as shown in Figure *1*. Random forest approaches network traffic classification with specific sensitivity to high-dimensional datasets and the discovery of complex relationships between features.

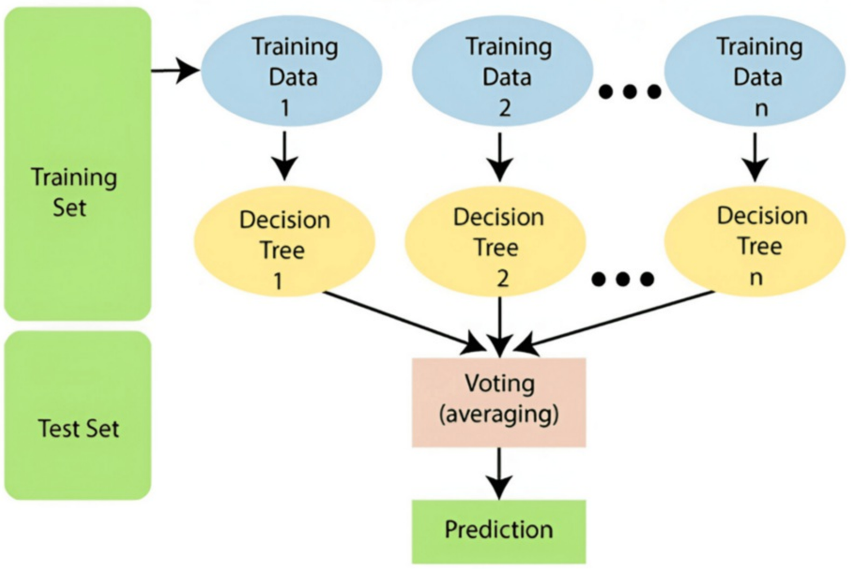

FIGURE 1: Random forest

Source: [25]

Random forest works by randomly choosing subsets of features and data points to construct each tree, thus diversifying the different trees and preventing the system from overfitting. This property is particularly helpful for network traffic classification, as the data may have complex patterns and interactions. The model can classify both categorical and continuous data whenever normal and malicious traffic are properly differentiated.

In addition, random forest provides feature importance scores, which may enable ranking factors in traffic classification and hence directly contribute to both improving the interpretability of the model and overall performance. Its simplicity, efficiency, and high accuracy make random forest a powerful tool for network traffic analysis and classification.

One of the most popular supervised learning algorithms for classification, the decision tree is one of the reasons why it performs well on network traffic classification problems due to its simplicity and interpretability. The decision tree algorithm works on the principle of recursively splitting the dataset into subsets based on feature values. Finally, it produces a tree structure where each node is a representative of a feature, and the branches represent the decision rules.

Decision trees are helpful in distinguishing between different forms of traffic, including benign and malicious traffic in the context of network traffic classification. Its ability to handle both categorical and continuous features makes it best suited for modeling complex decision boundaries in data and thus serves well in identifying the patterns behind network traffic flows.

The interpretability feature of decision trees is a strength, as it clearly shows an understanding of the decision-making process through the interpretation of the tree structure. However, the potential of decision trees to overfit, especially when the tree gets complex, leads to techniques such as pruning or using ensemble methods like random forest to improve generalization in traffic classification tasks. k-NN is one of the most intuitive as well as the simplest supervised learning algorithms widely applied to classification problems. For example, network traffic can be classified. The rule works based on choosing 'k' closest data points (neighbors) in the feature space to the point in question and assigning the most common class among those neighbors. The non-parametric nature of k-NN makes it very useful in traffic classification. Non-parametric means that it does not assume an underlying distribution for the data, which is helpful when dealing with diverse network traffic patterns.

In network traffic classification, the performance of k-NN is basically dominated by the chosen value 'k' and distance metric used, for example, Euclidean or Manhattan distance. Given that network traffic can be quite varied, k-NN can do well in classifying traffic if it has realized similarities in the features such as sizes, times, or characteristics of flow. However, this incurs increased computational cost for bigger datasets, where optimizations may be required for real-time classification applications.

Deep learning-based models

CNN architecture would be a good fit for spatial patterns that may exist in the data; hence, it would suit handling either image or sequential datasets. For network traffic classification, these CNNs can learn automatically features pertinent to the traffic flow - essential features like packet size, flow directions, and type of protocol. Figure *2* displays the CNN model architecture, which composes convolutional layers and subsequent max pooling layers that extract hierarchical features while reducing dimensionality. Furthermore, to enhance model generalization and prevent overfitting, batch normalization and dropout layers have been implemented.

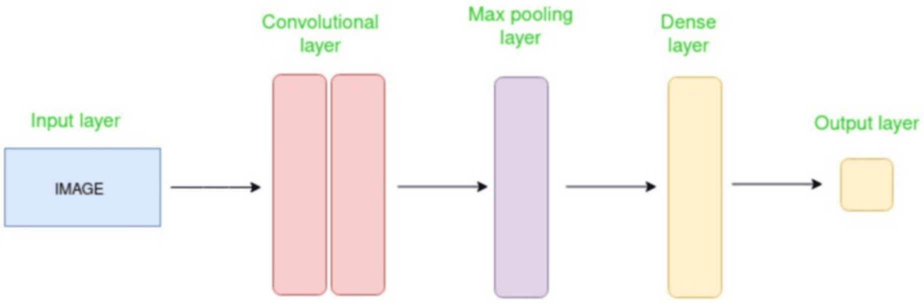

FIGURE 2: CNN architecture

Source: [26]

CNN, convolutional neural networks

RNN captures temporal dependencies in sequential data. Network traffic is usually a time-series phenomenon; thus, the models that contain RNNs tend to work very well for capturing traffic patterns over time. As shown in Figure *3*, the RNN architecture contains recurrent layers - for example, LSTM or gated recurrent unit layers that can capture long-range dependencies without suffering from vanishing gradients. This captures the complex temporal patterns in network traffic data because an RNN learns to process such data in a sequential manner and classify it into various types.

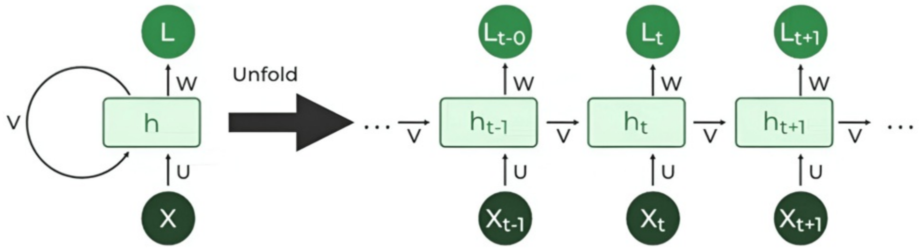

FIGURE 3: RNN architecture

Source: [27]

RNN, recurrent neural networks

The combination model, combining CNN and RNN, as demonstrated in Figure *4*, is a union of two strong architectures with great merging ability to handle complex tasks on data, particularly sequential ones. This hybrid approach exploits the strengths of CNNs that are noted in Figure *5*, depicting feature extraction capabilities. The sequential modeling capabilities of RNNs are noted in Figure *6*. In other words, by integrating the hierarchical feature extraction characteristic with RNN's capability for temporal dependency modeling, a hybrid model that can aptly analyze input data containing both spatial and temporal patterns is realized. In this model, the CNN component acts as a first processing layer for the extraction of main spatial features from the data, and RNN processes these features over time, thereby catching long-range relationships and temporal dependencies. The hybrid model has advantages by integrating CNNs and RNNs; it utilizes information from both spatial and temporal domains, providing a more robust and comprehensive representation of input data. Standard training techniques, like back propagation, gradient descent, and evaluation of performance, are used to make this model effective in applications for image recognition, natural language processing, time series analysis, among others.

The LSTM model plays a highly important role in capturing complex temporal patterns, which are inherent within encrypted flows. This is very important for traffic classification tasks. The first description of the LSTM model was done by Hochreiter and Schmidhuber [28], making this model effectively produce the right arrangements for sequentially showing long-term dependencies, which is just perfect for this application. Unlike simple RNNs, LSTM uses memory cells with the help of gating mechanisms. Using this feature, information can be stored or forgotten selectively based on the need for computation in time. This is an extremely important feature when it comes to detecting small temporal dependencies in encrypted traffic data. The specific architectures of LSTM help eliminate many of the common problems that tend to occur with simple RNNs, such as the problem of vanishing gradients. The use of back propagation through time for fine-tuning the parameters makes the LSTM capture more efficiently the temporal dynamics of the data. As a consequence, the overall hybrid architecture will boost its performance using the LSTM model, while correctly extracting the temporal features responsible for traffic classification with higher precision.

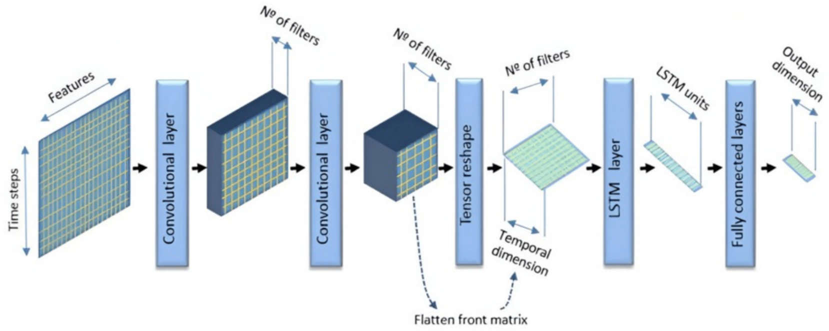

FIGURE 4: Combination of CNN and RNN

Source: [29]

CNN, convolutional neural networks; RNN, recurrent neural networks; LSTM, long short-term memory

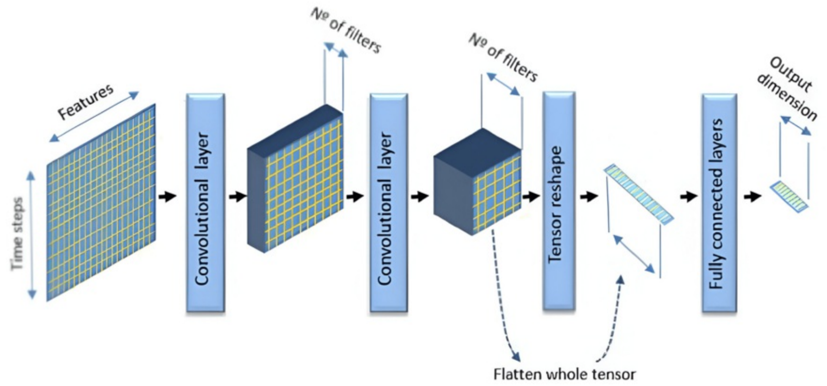

FIGURE 5: Feature extraction capabilities of CNN model

Source: [29]

CNN, convolutional neural networks

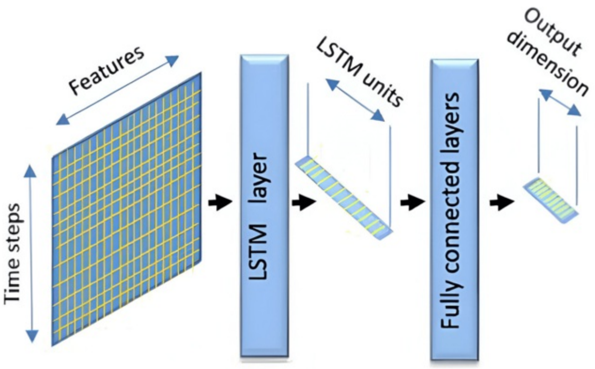

FIGURE 6: Sequential modeling capabilities of RNN model

Source: [29]

RNN, recurrent neural networks; LSTM, long short-term memory

Results

Architecture and experimental setup

The structure of the network traffic classification in Figure 7 is divided into different steps, starting from choosing and preprocessing the chosen dataset. For this work, there are several datasets. In this research, three different datasets are used: UNSW-NB15, ISCXTor2016, and finally a phishing dataset, indicating the availability of labeled instances of normal and malicious traffic for the classification model to differentiate among various network activities [30-32]. Thanks to ISCXTor2016, researchers make realistic Tor network traffic important for advanced cybersecurity research, particularly within network traffic analysis and intrusion detection. Before model training, significant preprocessing is done on the dataset. These involve changing missing values in the data, normalizing the data, feature selection to avoid redundancy, and better performance of the model. Categorical labels are converted into numerical values that can be accepted by algorithms for machine learning as well as deep learning. After splitting the dataset, it is divided into two sets: the training set and the testing set.

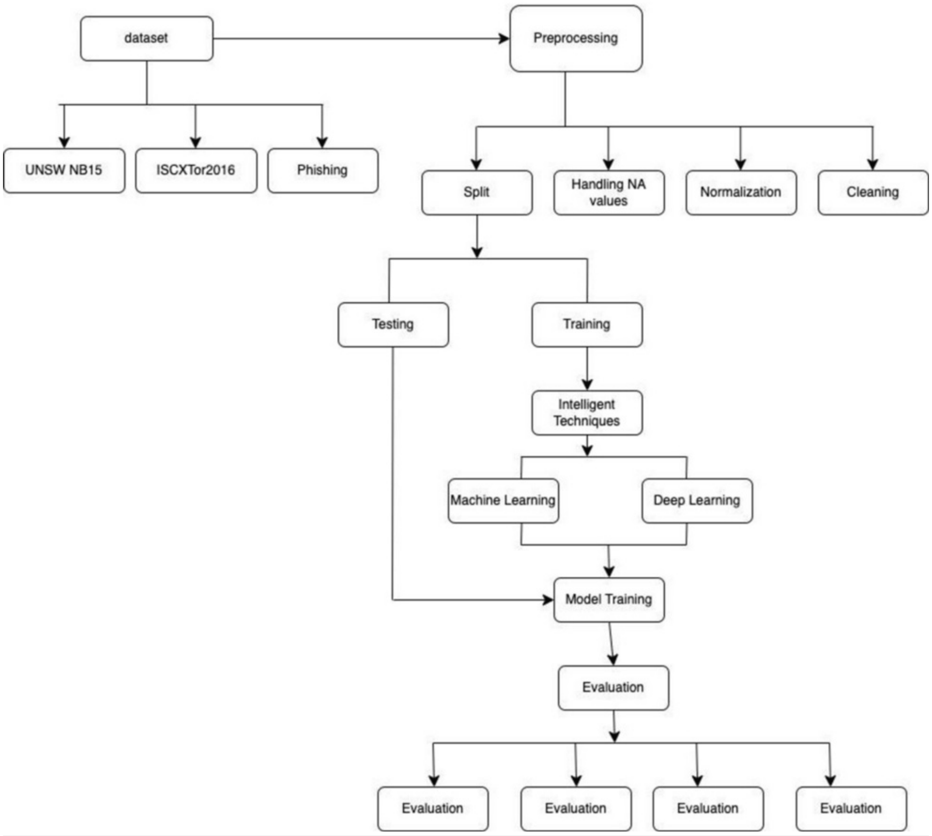

FIGURE 7: Proposed architecture

The architecture also manifests both machine learning and deep learning models, that appear as a part of the "Intelligent Techniques" layer. These latter models are trained using a systematic process involving several epochs; during these epochs, validation is conducted on training to monitor it at preventing overfitting. Finally, with the models trained appropriately, evaluation is conducted using appropriate metrics as regards the assessment of the classification accuracy as well as the generalization abilities.

Data cleaning is one of the necessary steps that occur in the preprocessing of the dataset. Duplicates should be removed because they might introduce bias and may even distort the learning process. Next, irrelevant data points - like a list of entries with missing or erroneous labels, or those not contributing to classification - will be found and removed. This means that the dataset on which the model is to be trained is both valid and relevant, bringing an improvement in the overall performance of the model.

After data cleaning, it is a strict necessity that all features are on some comparable scale. This is through Min-Max scaling, which is a technique to transform the value of each feature to an interval, typically between 0 and 1. By scaling the features, all features are normalized; no single feature overshadows the others due to scale differences that may otherwise cause unbalanced and ineffective modeling. It is particularly relevant when applying algorithms that are sensitive to the magnitude of input features, such as SVM and k-NN.

The processing missing values serve as an important operation in maintaining the integrity of the dataset. In this step, missing values within features are imputed, meaning they are replaced by appropriate substitutes. Another very common imputation method is the one that uses the median value of the relevant feature. This is due to its slight sensitivity to outliers as a comparison point with the mean. In this manner, the resulting dataset will remain complete, and all features are included for model training purposes, thus avoiding exclusion of potentially precious data points.

To evaluate the performances of these machine learning models, this set of data has to be split into two sets, namely, training set and testing set. The split I used here is a 70-30 split, which means that 70% is the training set used as training data to train the model, while the remaining 30% is the testing set. This split also assures that the bulk of the data will be utilized as training for the model while retaining an adequate proportion of the data to be used in its actual assessment for generalization performance. It is also important that the class distribution of training and testing sets be realistically representative of the overall dataset, lest favor the model inordinately on any given evaluation for a particular class.

UNSW-NB15 dataset has 56,551 rows and 45 columns initially, which further reduced during data preprocessing phase. Figure 8 is a bar plot showing the relationship between the category of attack and service it used as it can be that the generic attack has DNS services. In Figure 9, the count plot reveals that the number of samples in the "normal" category ranges between 30,000 and 35,000, while the number of worms is less than 5,000.

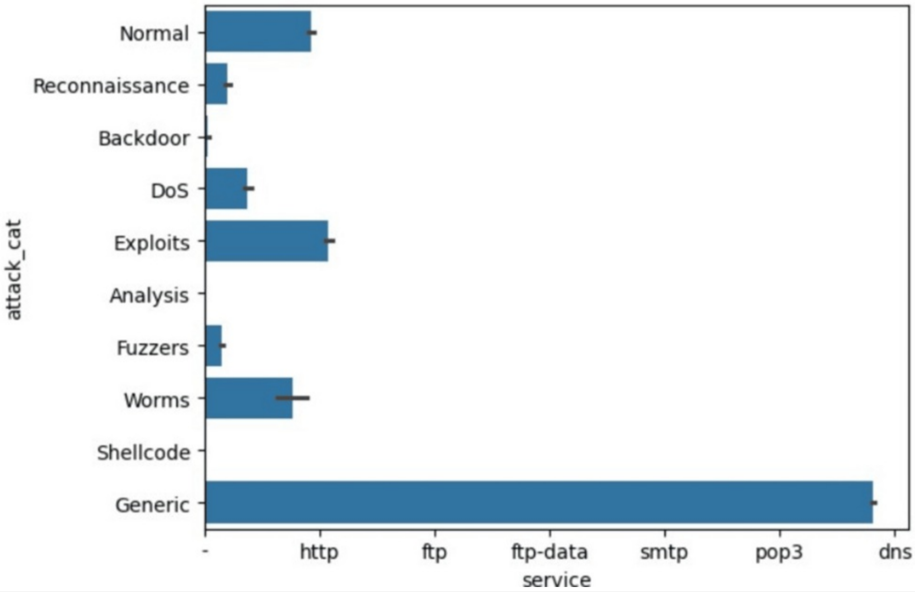

FIGURE 8: Bar plot of attack category vs. service

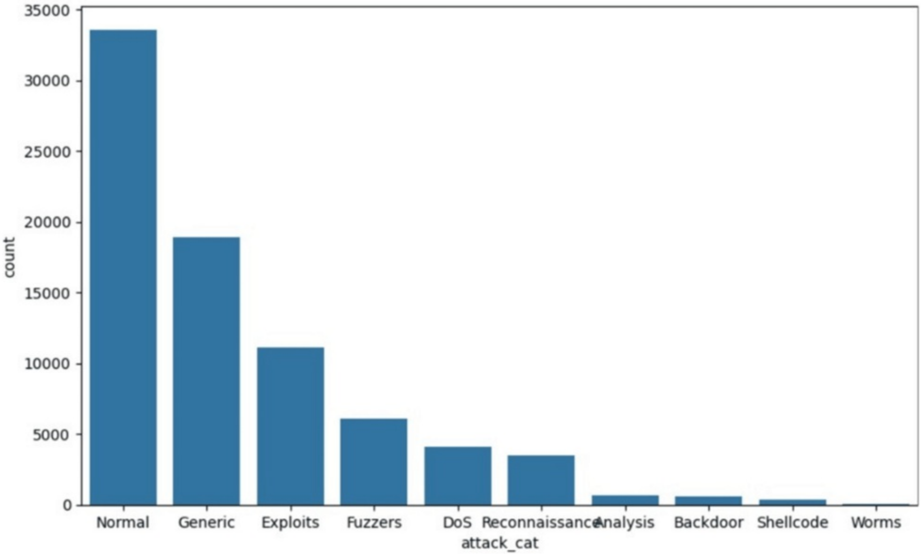

FIGURE 9: Count plot of target variable in UNSW-NB15 dataset

Figure 10 shows that spam records have the largest query length compared to benign, malware, phishing, and defacement.

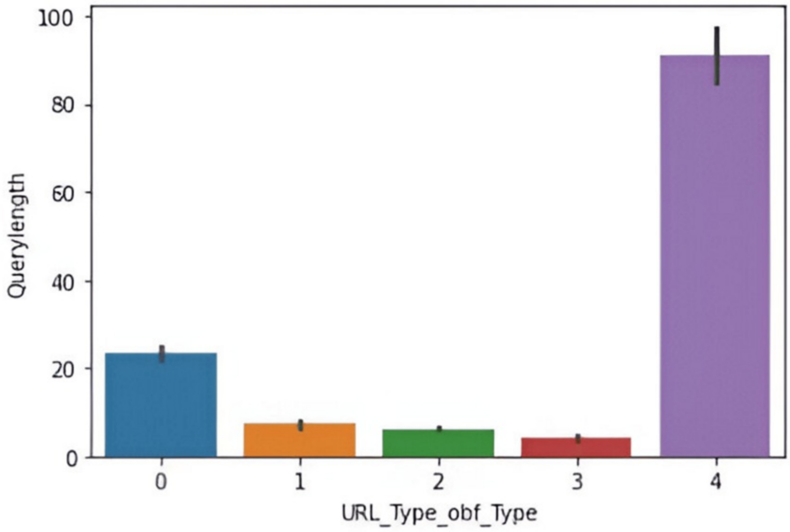

FIGURE 10: Count plot of URL type vs. query length

URL, uniform resource locator

Careful fine-tuning of the hyperparameters of these machine learning and deep learning models enhanced the performance of the models using optimization techniques combined with grid search, random search, and cross-validation. They systematically explore different combinations of hyperparameter settings to find suitable ones which can bring the highest accuracy and efficiency of the model. The two methods are grid search, which tries all possible combinations of hyperparameters within a predefined range, and random search, which conducts it randomly but sometimes can reach the result in a much faster way if dealing with a large search space. Cross-validation ensures the model is tested on different subsets of the data, thus giving a better estimate of how the model will perform.

Several algorithms differ by their potential and weaknesses. The choice is mostly based on specific tasks, available data, and the trade-offs between accuracy, interpretability, and computational complexity. In the current work, both types of models were considered: machine learning models (SVM, random forest, decision trees, and KNN) and deep learning models (CNN and RNN). The usage of preprocessing techniques, for instance, feature engineering and normalization by the data, improved all models significantly and enhanced the classification of traffic flow. The deep learning models were trained using the Adam optimizer in conjunction with the binary cross-entropy loss function. Training is performed over several epochs with early stopping to prevent overfitting. Model checkpoints are used to periodically save and track the training of a model, making it possible to fine-tune or retrain if needed. As such, we have been able to incorporate machine learning and deep learning techniques with productive preprocessing to yield a general yet balanced approach to traffic classification.

Other than hyperparameter tuning, the linear relationship strength between pairs of features was measured using Pearson's correlation coefficient. All features with a high correlation coefficient were discovered for possible elimination. Highly correlated features must be redundant; the information they provide can increase model complexity without improving accuracy and might even make the model worse when suffering from multicollinearity. Eliminating these redundant features simplifies the model and improves its efficiency and accuracy of prediction without sacrificing useful information.

Network traffic data were obtained as a combination of UNSW-NB15 dataset, ISCXTor2016 dataset, and phishing dataset. UNSW-NB15 dataset is a publicly available dataset widely used for network intrusion detection and classification tasks [30]. It provides a comprehensive collection of network traffic instances captured in a controlled environment, allowing for the analysis and classification of various network activities. The dataset contains labeled instances of network traffic, including both benign and malicious activities. The dataset contains features like srcip, sport, dstip, proto, and others, and the description of these features is given in Table *1*. It encompasses diverse network traffic scenarios, such as normal user activities, port scans, denial-of-service (DoS) attacks, and botnet activities. Each instance in the dataset is labeled with the corresponding network traffic class, indicating whether it belongs to a normal class or a specific type of attack class. UNSW-NB15 dataset has 56,551 rows and 45 columns initially, which is further reduced during the data preprocessing phase. The dataset contains a combination of categorical and numeric values.

Features	Description
srcip	Source IP address from where the network packet comes.
sport	Source port used in communication.
dstip	Destination IP address to which the network packet is sent.
dsport	Destination port used in the dataset.
proto	Network protocol used in the communication.
dur	Total duration in seconds.
dbytes	Received bytes at destination IP address.
nsttl	TTL value of source IP address.
nsloss	Lost packets of source IP.
ndloss	Lost packets of destination IP.
service	Service on which the destination is running (e.g., http, ftp, ssh, etc.).
nsload	Number of source bits per second.
ct ftp cmd	Number of FTP commands sent during this session.
ct srv src	Connections to the same service in the time window originated by source IP.
Attack cat	Type of attack (e.g., DOS, Exploit, Fuzzers, etc.).

TABLE 1: Description of features in UNSW-NB15 dataset

IP, internet protocol; TTL, time to live; FTP, file transfer protocol

The ISCXTor2016 dataset is a valuable resource widely used in cybersecurity research, particularly in the field of network traffic analysis and intrusion detection [31]. It comprises network traffic data collected from a realistic Tor network environment, specifically designed to simulate real-world scenarios involving Tor usage. The dataset offers a diverse range of network traffic samples, including both normal and malicious activities, thus enabling researchers to develop and evaluate intrusion detection systems tailored to detect Tor-related threats effectively. ISCXTor2016 provides labeled network traffic instances, facilitating supervised learning approaches for traffic classification tasks. ISCXTor2016 data contain features like flow duration, total Fwds/Bwds, TotLen Fwd/Bwd packets, etc., given in Table *2*. This dataset is instrumental in fostering research aimed at enhancing network security, understanding Tor network behavior, and developing robust defense mechanisms against malicious activities within the Tor network.

Features	Description
Flow Duration	Total duration of the flow in microseconds
Tot Fwd/Bwd Pkts	Total number of packets send in forward/backward direction
TotLen Fwd/Bwd Pkts	Total length of packets in forward/backward direction
Flow Byts/Pkts	Number of bytes/packets per second in the flow
Fwd/Bwd IAT Mean	Mean inter-arrival time between forward/backward packets
Fwd/Bwd PSH flags	Number of times the PSH flag is sent in packets moving forward/backward
Down/up Ratio	Ratio of bytes in the forward direction to those in the backward direction
Subflow Bwd/Fwd Byts	Number of subflow bytes in the forward/backward direction
Label	Target label that identifies the type of traffic

TABLE 2: Description of features in ISCXTor2016 dataset

Another dataset used is phishing dataset [32]. It is used to train machine learning models that can detect potential phishing attempts, enhancing cybersecurity measures. This dataset is used to classify URLs into benign and phishing. The various features present in this dataset are features related to count of letter, frequency of characters in the URL calculated in the form of letters, tokens, symbol, etc., and are presented in Table 3. These characters are categorized and counted from these components of URLs.

Features	Description
Symbol Count in Domain	The frequency of specific delimiters such as ://, ., /, ?, =, ;, (,), and +
Domain Token Count	This feature assesses the number of tokens derived from the URL string
Query Digit Count	This represents the total number of digits found in the query segment of the URL
Top-Level Domain Usage	Phishing URLs sometimes utilize multiple top-level domains within a single domain n
Number Rate in Various URL Parts	This metric calculates the proportion of numbers within different parts of the URL
Features Related to Length	The overall length of a URL, including specific parts like the domain and filename
Letter-Digit-Letter Sequence Detection	Phishing URLs often use patterns that intermix letters and digits (e.g., ldl patterns)

TABLE 3: Description of features in phishing dataset

URL, uniform resource locator

To evaluate the effectiveness of our classification methods, this research has utilized four commonly adopted metrics: precision (Pr) or positive predictive value, recall (Rc) or sensitivity, accuracy, and F1-score. Precision measures the ratio of correctly identified positive instances (true positives) to the total instances classified as positive, both correctly and incorrectly.

Precision is calculated as

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (1)$$

Recall, on the other hand, quantifies the proportion of actual positive instances that are correctly identified by the model.

Recall is computed as

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (2)$$

where false negatives represent the positive instances incorrectly classified as negative.

Accuracy is correctly classified out of total instances

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3)$$

F1-score is the harmonic mean of precision and recall.

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (4)$$

Figure *11* provides a graphical comparison of performance metrics - accuracy, precision, recall, and F1-score - for both machine learning and deep learning models applied to the UNSW-NB15 dataset. This dataset, designed for network intrusion detection, highlights the superior performance of models such as CNN, RNN, and random forest. These models demonstrate remarkable accuracy, recall, and F1-scores, as outlined in Table *4*, showcasing their robustness in detecting complex network patterns.

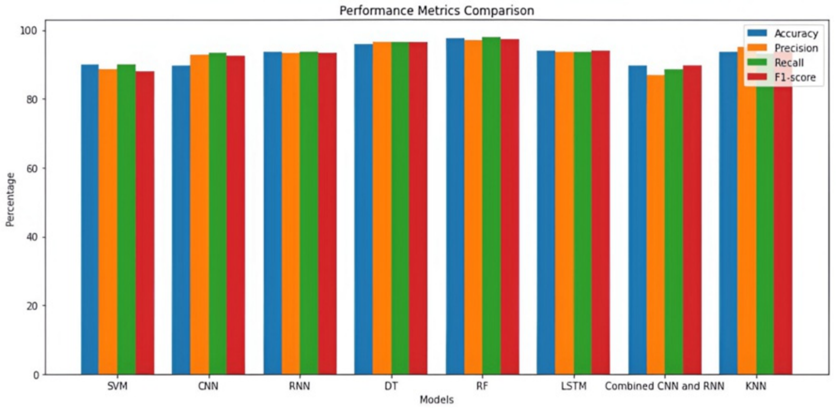

FIGURE 11: Performance metrics over UNSW-NB15 dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; DT, decision tree; RF, random forest; LSTM, long short-term memory; KNN, k-nearest neighbors

Models	Accuracy	Precision	Recall	F1-score
SVM	88.126	88.126	88.126	88.126
CNN	93.29	92.92	93.29	92.62
RNN	93.65	93.43	93.65	93.65
Decision tree	93.78	94.00	94.00	94.00
Random forest	91.99	97.13	90.61	93.76
LSTM	75.00	80.93	72.90	76.70
Combined CNN and RNN	88.50	87.32	85.35	86.32
KNN	81.19	82.00	82.00	82.00

TABLE 4: Performance metrics over UNSW-NB15 dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; LSTM, long short-term memory; KNN, k-nearest neighbors

Table 5 summarizes the performance metrics - accuracy, precision, recall, and F1-score - of machine learning and deep learning models applied to the ISCXTor2016 dataset. The models include SVM, CNN, RNN, the combination of CNN and RNN, decision tree, random forest, and LSTM. Correspondingly, Figure 12 offers a visual representation of these metrics, illustrating the comparative strengths of each model.

Models	Accuracy	Precision	Recall	F1-score
SVM	89.96	88.49	89.96	87.87
CNN	93.29	92.92	93.29	92.62
RNN	93.65	93.43	93.65	93.65
Decision tree	93.78	94.00	94.00	94.00
Random forest	95.20	97.00	94.00	95.47
LSTM	93.70	93.59	93.51	93.59
Combined CNN and RNN	89.96	88.49	89.96	87.97
KNN	81.19	82.00	82.00	82.00

TABLE 5: Performance metrics over ISCXTor2016 dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; LSTM, long short-term memory; KNN, k-nearest neighbors

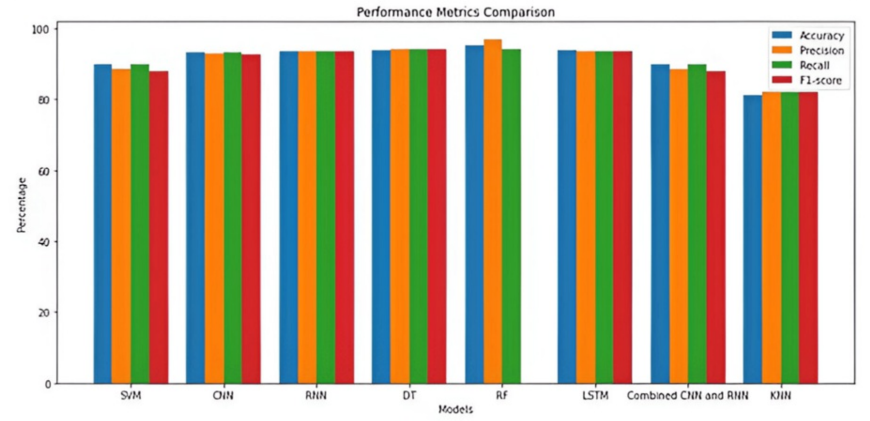

FIGURE 12: Performance metrics over ISCXTor2016 dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; DT, decision tree; RF, random forest; LSTM, long short-term memory; KNN, k-nearest neighbors

Table 6 illustrates the various performance metrics of machine learning and deep learning models applied to the phishing dataset. The models used for this evaluation are SVM, CNN, RNN, CNN+RNN, decision tree, random forest, LSTM, and k-NN. Figure 13 graphically demonstrates the comparative results of these models in detecting phishing attacks.

Models	Accuracy	Precision	Recall	F1-score
SVM	89.96	88.49	89.96	87.87
CNN	89.83	92.92	93.29	92.62
RNN	93.59	93.34	93.59	93.43
Decision tree	95.83	96.40	96.60	96.40
Random forest	97.77	97.00	98.00	97.49
LSTM	94.00	93.70	93.78	94.00
Combined CNN and RNN	86.89	88.63	89.83	88.30
KNN	93.61	95.00	93.00	93.98

TABLE 6: Performance metrics over phishing dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; LSTM, long short-term memory; KNN, k-nearest neighbors

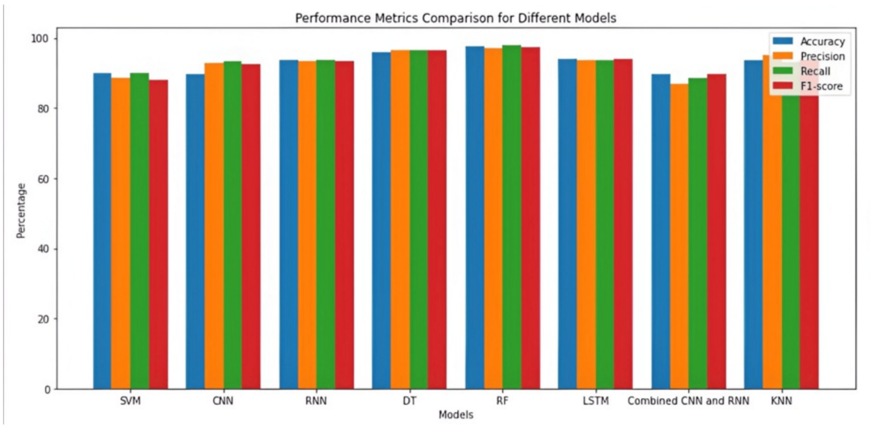

FIGURE 13: Performance metrics over phishing dataset

SVM, support vector machines; CNN, convolutional neural networks; RNN, recurrent neural networks; DT, decision tree; RF, random forest; LSTM, long short-term memory; KNN, k-nearest neighbors

Discussion

Deep learning models consistently outperformed most machine learning models in network traffic classification tasks. The superior performance of CNN and RNN models can be attributed to their ability to automatically learn intricate patterns and hierarchical features from raw network traffic data. These models excel at capturing the complex temporal dependencies inherent in network traffic, which is crucial for accurate classification. Notably, the RNN and CNN models from deep learning, along with random forest and decision tree models from machine learning, exhibited slightly better performance compared to other models across all evaluation metrics. This highlights the importance of leveraging sequential information in network traffic classification, as it allows the models to better understand the underlying patterns and anomalies present in the data. These results suggest that deep learning models, especially those capable of handling sequential data, are better suited to handle the complexities of modern network environments

Conclusions

This study has several noteworthy limitations. The datasets used, UNSW-NB15 and ISCXTor2016, while valuable for research purposes, may not fully capture the diversity of real-world network traffic across all scenarios. Additionally, the computational cost associated with deep learning models poses a significant challenge, particularly when handling large-scale datasets that demand substantial computational power. Despite these limitations, the findings highlight the practical relevance of deep learning models - especially

CNNs and RNNs - in network traffic classification. These models excel at identifying complex patterns in encrypted data and can be integrated into real-time traffic monitoring systems to improve the detection of sophisticated cyber threats, such as phishing and advanced cyberattacks, which traditional detection methods might miss. A combination of machine learning and deep learning models offers further performance enhancements in dynamic traffic environments with highly variable patterns.

The comprehensive evaluation demonstrates that deep learning models consistently surpass most traditional machine learning models in classification accuracy. The superior results of CNN and RNN models stem from their ability to automatically learn detailed patterns and hierarchical features from raw network traffic data. Among the evaluated models, RNN, CNN, random forest, and decision tree models exhibited slightly better performance across all metrics, emphasizing the importance of utilizing sequential information for network traffic classification. These results highlight the growing potential of deep learning in enhancing network security. Future research could explore advanced neural architectures, incorporate attention mechanisms, and apply transfer learning to further refine classification performance in practical applications. Overall, this study advances the field of network security by demonstrating the effective use of deep learning techniques for analyzing and classifying network traffic.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Acquisition, analysis, or interpretation of data: Aswani Kumar Cherukuri, Shudhamati Mali, Mansi Gujral

Drafting of the manuscript: Aswani Kumar Cherukuri, Shudhamati Mali, Mansi Gujral

Critical review of the manuscript for important intellectual content: Aswani Kumar Cherukuri

Supervision: Aswani Kumar Cherukuri

Concept and design: Shudhamati Mali

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Getman AI, Ikonnikova MK: A survey of network traffic classification methods using machine learning . Programming and Computer Software. 2022, 48:413-423. 10.1134/s0361768822070052
2. Izadi S, Ahmadi M, Rajabzadeh A: Network traffic classification using deep learning networks and Bayesian data fusion. Journal of Network and Systems Management. 2022, 30:25. 10.1007/s10922-021-09639-z
3. Shafiq M, Yu X, Laghari AA, Yao L, Karn NK, Abdessamia F: Network traffic classification techniques and comparative analysis using machine learning algorithms. 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China. 2016, 2451-2455. 10.1109/CompComm.2016.7925139
4. Singh K, Agrawal S: Comparative analysis of five machine learning algorithms for IP traffic classification . 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, India. 2011, 33-38. 10.1109/ETNCC.2011.5958481
5. Cherukuri AK, Ikram ST, Li G, Liu X: Encrypted Network Traffic Analysis. Springer, Cham; 2024. 10.1007/978-3-031-62909-9
6. Rezaei S, Liu X: Deep learning for encrypted traffic classification: An overview . IEEE Communications Magazine. 2019, 57:76-81. 10.1109/mcom.2019.1800819
7. Dakic P, Zivkovic M, Jovanovic L, Bacanin N, Antonijevic M, Kaljevic J, Simic V: Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles. Scientific Reports. 2024, 14:22884. 10.1038/s41598-024-73932-5
8. Khafaga DS, Karim FK, Abdelhamid AA, et al.: Voting classifier and metaheuristic optimization for network intrusion detection. Computers, Materials & Continua. 2023, 74:3183-3198. 10.32604/cmc.2023.033513
9. Savanović N, Toskovic A, Petrovic A, et al.: Intrusion detection in healthcare 4.0 Internet of Things systems via metaheuristics optimized machine learning. Sustainability. 2023, 15:12563. 10.3390/su151612563
10. Lim HK, Kim JB, Heo JS, Kim K, Hong YG, Han YH: Packet-based network traffic classification using deep learning. International Conference on Artificial Intelligence in Information and Communication (ICAIIC)),

Okinawa, Japan. 2019, 46-51. 10.1109/ICAIIC.2019.8669045

11. Özdel S, Damla Ateş P, Ateş Ç, Koca M, Anarım E: Network traffic classification with flow based approach. 30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Türkiye. 2022, 1-4. 10.1109/SIU55565.2022.9864760
12. Zuev D, Moore AW: Traffic classification using a statistical approach. Passive and Active Network Measurement. Dovrolis C (ed): Springer, Berlin, Heidelberg; 2005. 321-324. 10.1007/978-3-540-31966-5_25
13. Auld T, Moore AW, Gull SF: Bayesian neural networks for Internet traffic classification. IEEE Transactions on Neural Networks. 2007, 18:223-239. 10.1109/tnn.2006.883010
14. Nguyen TTT, Armitage G: A survey of techniques for internet traffic classification using machine learning . IEEE Communications Surveys & Tutorials. 2008, 10:56-76. 10.1109/surv.2008.080406
15. Williams N, Zander S, Armitage G: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review. 2006, 36:5-16. 10.1145/1163593.1163596
16. Yang J: The application of deep learning for network traffic classification. Highlights in Science, Engineering and Technology. 2023, 39:979-984. 10.54097/hset.v39i.6689
17. AlEroud A, Karabatis G: Using contextual information to identify cyber-attacks. Information Fusion for Cyber-Security Analytics. Studies in Computational Intelligence. Springer, Cham; 2017. 691:1-16. 10.1007/978-3-319-44257-0_1
18. Aouedi O, Piamrat K, Parrein B: Ensemble-based deep learning model for network traffic classification. IEEE Transactions on Network and Service Management. 2022, 19:4124-4135. 10.1109/tnsm.2022.3193748
19. Xue Y, Wang D, Zhang L: Traffic classification: Issues and challenges. International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, USA. 2013, 27:545-549. 10.1109/ICCNC.2013.6504144
20. Jovanovic L, Jovanovic D, Antonijevic M, Nikolic B, Bacanin N, Zivkovic M, Strumberger I: Improving phishing website detection using a hybrid two-level framework for feature selection and XGBoost tuning. Journal of Web Engineering. 2023, 22:543-574. 10.13052/jwe1540-9589.2237
21. Rachmawati SM, Kim D-S, Lee J-M: Machine learning algorithm in network traffic classification. International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea. 2019, 4:1010-1013. 10.1109/ICTC52510.2021.9620746
22. Martínez Torres J, Iglesias Comesaña C, García-Nieto PJ: Review: machine learning techniques applied to cybersecurity. International Journal of Machine Learning and Cybernetics. 2019, 10:2823-2836. 10.1007/s13042-018-00906-1
23. Zivkovic M, Tair M, Venkatachalam K, Bacanin N, Hubálovský Š, Trojovský P: Novel hybrid firefly algorithm: an application to enhance XGBoost tuning for intrusion detection classification. PeerJ Computer Science. 2022, 8:e956. 10.7717/peerj-cs.956
24. Li J, Pan Z: Network traffic classification based on deep learning. KSII Transactions on Internet and Information Systems. 2018, 14:4246-4267. 10.3837/tiis.2020.11.001
25. https://www.javatpoint.com/machine-learning-random-forest-algorithm.
26. https://www.geeksforgeeks.org/introduction-convolution-neural-network/.
27. https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.
28. Hochreiter S, Schmidhuber J: Long short-term memory. Neural Computation. 1997, 9:1735-1780. 10.1162/neco.1997.9.8.1735
29. Lopez-Martin M, Carro B, Sanchez-Esguevillas A, Lloret J: Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. IEEE Access. 2017, 5:18042-18050. 10.1109/access.2017.2747560
30. Moustafa N, Slay J: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia. 2015, 1-6. 10.1109/MilCIS.2015.7348942
31. Habibi Lashkari A, Draper Gil G, Mamun MSI, Ghorbani AA: Characterization of Tor traffic using time based features. Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP. 2017, 1:253-262. 10.5220/0006105602530262
32. Mamun MSI, Rathore MA, Lashkari AH, Stakhanova N, Ghorbani AA: Detecting malicious URLs using lexical analysis. Network and System Security. Chen J, Piuri V, Su C, Yung M (ed): Springer, Cham; 2016. 9955:467-482. 10.1007/978-3-319-46298-1_30