CLASSIFICATION OF NETWORK TRAFFIC USING MACHINE LEARNING MODELS ON THE NETML DATASET

Mezati Messaoud

Department of Computer , Kasdi Merbah University, Ouargla, Algeria

ABSTRACT

Network traffic classification plays a critical role in cybersecurity, quality of service (QoS) management, and anomaly detection. Traditional rule-based classification methods struggle with the increasing complexity and volume of network traffic, necessitating the adoption of machine learning (ML) techniques. In this study, we explore the effectiveness of ML models in classifying network traffic using the NetML dataset, a benchmark dataset that captures diverse traffic patterns, including benign and malicious activities. We preprocess the dataset by applying feature selection, normalization, and data balancing techniques to optimize model performance. Several ML models, including traditional classifiers such as Random Forest (RF), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), as well as deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, are trained and evaluated. Model performance is assessed using accuracy, precision, recall, F1-score, and AUC-ROC metrics. Experimental results demonstrate that deep learning models, particularly LSTM networks, achieve superior performance in capturing temporal dependencies in network traffic, significantly outperforming traditional classifiers. Our results indicate that LSTM, GRU, and CNN models all achieved an accuracy of 92.26%, highlighting their effectiveness in network traffic classification. Additionally, feature selection techniques improved computational efficiency without compromising classification performance. However, confusion matrix analysis revealed that the models tend to predict the most frequent class, leading to potential bias and lower accuracy for minority classes. The study also highlights the presence of high values in the confusion matrices, exceeding 70,000 in some cases, indicating dataset imbalance and model bias toward dominant classes. Despite achieving high accuracy, misclassification challenges persist, particularly in identifying encrypted traffic and polymorphic attacks. Transformer-based models demonstrated resilience to adversarial modifications but required significantly higher computational resources. Future work should explore adversarial training, self-supervised learning, and hybrid CNN-LSTM architectures to enhance robustness against evolving cyber threats. Additionally, feature selection optimization and hyperparameter tuning can further refine classification performance, ensuring more reliable deployment in real-world cybersecurity applications.

KEYWORDS

Machine Learning, Network Traffic Classification, NetML Dataset, Deep Learning, Cybersecurity

1. INTRODUCTION

The rapid growth of the Internet and digital communications has resulted in an exponential surge in network traffic.[1]. As networks become more complex and data volumes grow, ensuring secure, efficient, and well-managed traffic flow has become a critical challenge[2]. Network traffic classification—the process of categorizing network flows based on their characteristics—is a fundamental technique in cybersecurity, anomaly detection, and Quality of Service (QoS) management[3]. Accurate classification helps detect cyber threats, optimize bandwidth allocation, and improve network performance[4]. Traditional classification methods, such as

Deep Packet Inspection (DPI) and rule-based approaches, have been widely used in this field[5]. However, these methods face increasing limitations due to the rise of encrypted traffic, evolving attack patterns, and the need for real-time processing in large-scale networks.

Machine learning (ML) has emerged as a powerful tool for network traffic classification, offering the ability to recognize complex patterns and adapt to new traffic behaviors without requiring deep packet inspection[6][7][8][9][10]. Unlike traditional approaches, ML models rely on statistical flow-based features, making them effective even when traffic is encrypted. Various ML techniques have been explored, ranging from conventional classifiers such as Random Forest (RF)[11] and Support Vector Machines (SVM)[12] to deep learning architectures like Convolutional Neural Networks (CNNs)[13] and Long Short-Term Memory (LSTM) networks[14]. Despite the advancements in ML-based classification, selecting the optimal model and feature set for real-world deployment remains a challenge. Many studies rely on outdated or limited datasets, making it difficult to benchmark new approaches effectively.

Although ML-based classification has shown promise, there are still open questions regarding the scalability, adaptability, and robustness of these models in dynamic network environments. Key challenges include:

- Identifying the most relevant features that contribute to accurate classification while minimizing computational overhead.
- Understanding the trade-offs between different ML architectures in terms of accuracy, efficiency, and real-time applicability.
- Evaluating model performance on diverse datasets that capture realistic network conditions, particularly those with a mix of benign and malicious traffic.
- The NetML dataset provides a comprehensive and up-to-date benchmark for addressing these challenges. However, existing studies have not fully explored its potential in comparing different ML techniques for network traffic classification.

In this study,We utilize the NetML dataset [14] to systematically assess various machine learning models for network traffic classification. We preprocess the dataset using feature selection and normalization techniques, then train and compare multiple ML models, including RF, SVM, CNN, and LSTM architectures. Our results highlight the effectiveness of deep learning approaches, particularly LSTM, in capturing temporal dependencies in network traffic. Additionally, we examine the impact of feature selection on classification performance and computational efficiency. These findings provide insights into the deployment of ML models for real-world cybersecurity applications, contributing to the development of more scalable and accurate traffic classification systems.

2. RELATED WORK

Traditional network traffic classification methods have relied on rule-based techniques such as Deep Packet Inspection (DPI) and port-based analysis[15][16][17]. While DPI provides high accuracy by examining packet payloads for predefined signatures, it is computationally expensive and ineffective for encrypted traffic. Similarly, port-based classification, which associates traffic types with well-known port numbers, has become unreliable due to dynamic port allocation and the widespread use of port obfuscation techniques. These limitations have driven the adoption of machine learning (ML) approaches, which analyze flow-based statistical features rather than packet contents, making them more adaptable to evolving network conditions.

Machine learning techniques for traffic classification range from traditional models, such as Random Forest (RF), Support Vector Machines (SVM), and Decision Trees (DT), to advanced

deep learning architectures, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Traditional ML models require manual feature selection and often struggle with capturing temporal dependencies in network flows. Deep learning models[18][19][20][21], on the other hand, can automatically learn hierarchical patterns from raw network data. CNNs are effective in recognizing spatial feature correlations, while LSTMs are well-suited for analyzing sequential dependencies in time-series network flows. However, deep learning approaches require significant computational resources and large, diverse datasets for effective training.

While the study evaluates ML-based approaches for network traffic classification, a direct comparison with existing methods is essential. Traditional traffic classification techniques, such as Deep Packet Inspection (DPI), rule-based filtering, and port-based analysis, have been widely used but face significant limitations, particularly when dealing with encrypted traffic. Machine learning-based classification has gained popularity due to its ability to analyze flow-based features rather than inspecting raw payloads. Traditional ML models such as Random Forest (RF), Support Vector Machines (SVM), and Decision Trees (DT) have been extensively studied, but they often require manual feature selection and fail to capture sequential dependencies in network traffic. Deep learning models, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, provide significant improvements by learning hierarchical and temporal patterns. Transformer-based models have emerged recently as a promising alternative, offering robustness against adversarial modifications. A comparative analysis with studies using datasets such as CICIDS 2017, UNSW-NB15, or ISCX VPN-NonVPN would further highlight the advantages of the NetML dataset and the deep learning models evaluated in this work.

Several benchmark datasets have been used to evaluate ML models for network traffic classification, each with its own strengths and limitations. The CICIDS 2017 and UNSW-NB15 datasets provide a variety of normal and malicious traffic samples but lack comprehensive real-world diversity and contain imbalanced attack distributions. The ISCX VPN-NonVPN dataset focuses on distinguishing VPN traffic but does not represent broader network threats. In contrast, the NetML dataset offers a more comprehensive and feature-rich traffic dataset, including both benign and malicious flows, making it a valuable resource for evaluating modern ML-based classification models. Despite its potential, NetML remains underutilized in network traffic classification research.

Existing research faces several challenges, including the lack of diverse and up-to-date datasets, inefficient feature selection processes, and the need for scalable ML models suitable for real-time classification. Many studies focus on either traditional ML models or deep learning approaches without systematically comparing them under uniform experimental conditions. Furthermore, while deep learning has shown promise, its real-world deployment feasibility remains an open question due to computational constraints. This study aims to bridge these gaps by utilizing the NetML dataset to systematically compare traditional ML classifiers and deep learning models, refine feature selection, and assess their performance for real-time network traffic classification in contemporary cybersecurity applications.

3. DATASET DESCRIPTION

3.1. Overview of the NetML Dataset

The NetML dataset is a benchmark dataset designed to support machine learning-based network traffic classification[14]. It provides a diverse collection of real-world network traffic, including

both normal and malicious flows, making it particularly useful for evaluating the performance of ML models. Unlike older datasets that primarily focus on specific types of cyber threats, NetML offers a broad range of traffic types, allowing for more comprehensive analysis. The dataset is structured to facilitate both supervised and unsupervised learning approaches, making it suitable for various classification tasks, including anomaly detection and intrusion detection.

3.2. Data Sources and Collection Methods

The dataset was generated from real-world network environments, capturing both legitimate and attack traffic from different sources. Network traffic was collected using packet capture tools, including Wireshark and tcpdump, which recorded raw packet-level information. The collected data underwent preprocessing to extract statistical flow features, reducing the reliance on deep packet inspection while ensuring compatibility with ML-based classification methods. The dataset includes a mixture of traffic types from various applications, including web browsing, file transfers, streaming, and botnet activities. The diversity of data sources ensures that the dataset reflects realistic traffic patterns, enhancing its applicability to cybersecurity and network management research.

3.3. Types of Network Traffic Classes

The NetML dataset includes both benign and malicious traffic classes, categorized based on behavioral patterns and known attack signatures. Benign traffic consists of normal user activities such as HTTP and HTTPS browsing, email communication, and video streaming. Malicious traffic includes various cyber threats, such as DDoS attacks, botnet activities, port scanning, and exploitation attempts. Each traffic class is labeled based on its characteristics, allowing researchers to train and evaluate ML models on different types of threats and normal activities. The dataset supports both binary classification (benign vs. malicious) and multi-class classification, where specific attack types can be identified.

3.4. Feature Description

The dataset provides a rich set of features extracted from packet-level and flow-level data. The NetML dataset comprises over 60 distinct features, categorized into multiple groups, including network attributes, packet-level features, statistical flow features, DNS features, HTTP features, TLS features, and session & ID features. Statistical flow features and TLS-related attributes form the majority, enabling models to analyze network behavior without relying on payload inspection. This categorization ensures that machine learning models can effectively classify network flows based on statistical patterns rather than inspecting packet payloads, making the approach scalable and privacy-preserving. Rather than relying on raw payloads, the NetML dataset includes statistical flow features, which are crucial for classifying encrypted and obfuscated traffic. Features include:

- Basic network attributes: Source/destination IP addresses, ports, and protocols.
- Packet-level features: Packet size, inter-arrival time, and duration.
- Statistical flow features: Mean, variance, and standard deviation of packet sizes, flow duration, and byte counts per session.
- Behavioral metrics: Connection frequency, burst rates, and anomaly scores.

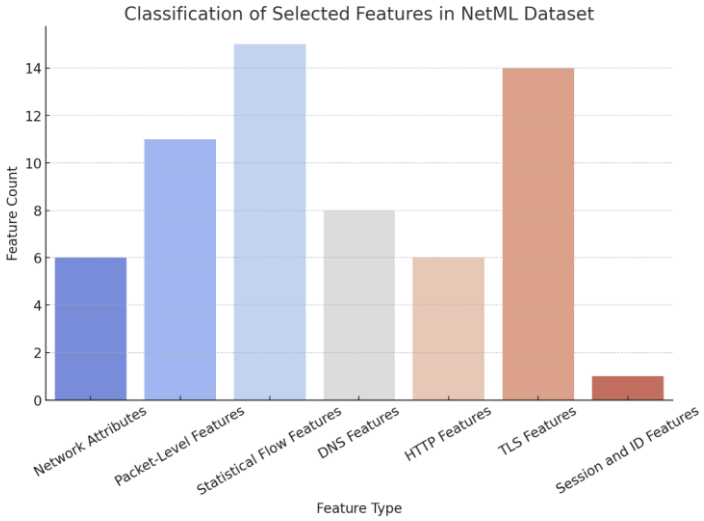

Figure 1. Feature Classification Distribution in NetML Dataset

This bar chart illustrates the distribution of classified features in the NetML dataset, categorizing them into Network Attributes, Packet-Level Features, Statistical Flow Features, DNS Features, HTTP Features, TLS Features, and Session & ID Features. The Statistical Flow Features and TLS Features categories contain the highest number of features, highlighting their importance in network behavior analysis and encrypted traffic monitoring. Conversely, Session and ID Features have the lowest count, indicating fewer attributes related to session tracking. The diverse feature distribution ensures that machine learning models trained on this dataset can effectively capture network behaviors, security threats, and performance metrics. Network Attributes and Packet-Level Features contribute to identifying traffic flow, while DNS and HTTP Features aid in detecting web-based anomalies. The prominence of TLS Features underscores the growing need for encrypted traffic analysis in cybersecurity. This classification supports the development of intrusion detection, anomaly detection, and performance monitoring systems, making the dataset valuable for modern network security applications.

Table 1. Detailed Feature Classification in NetML Dataset.

Feature Type	**Features**
Network Attributes	sa, pr, dst_port, src_port, da, dns_answer_ip
Packet-Level Features	rev_hdr_distinct, hdr_ccnt, bytes_in, rev_hdr_ccnt, hdr_mean, rev_hdr_bin_40, num_pkts_in, num_pkts_out, bytes_out, hdr_bin_40, hdr_distinct
Statistical Flow Features	intervals_ccnt, rev_pld_max, rev_pld_mean, pld_mean, rev_pld_ccnt, pld_bin_inf, rev_intervals_ccnt, rev_pld_distinct, pld_median, rev_pld_var, pld_distinct, pld_max, rev_pld_bin_128, time_length, pld_ccnt
DNS Features	dns_query_type, dns_query_class, dns_query_name_len, dns_query_name, dns_query_cnt, dns_answer_ip, dns_answer_ttl, dns_answer_cnt
HTTP Features	http_method, http_uri, http_host, http_code, http_content_len, http_content_type
TLS Features	tls_len, tls_key_exchange_len, tls_svr_ext_cnt, tls_svr_len, tls_svr_cs_cnt, tls_ext_cnt, tls_cnt, tls_svr_cs, tls_cs_cnt, tls_ext_types, tls_svr_key_exchange_len, tls_svr_ext_types, tls_svr_cnt, tls_cs
Session and ID Features	id

This table provides a structured categorization of selected features from the NetML dataset, grouping them into different feature types based on their roles in network traffic analysis. Each row represents a feature type, and the corresponding column lists the specific features that belong to that category.

- Network Attributes: Includes features related to network-level identifiers such as source and destination IP addresses, ports, and protocol types. These features help in identifying traffic sources and destinations.
- Packet-Level Features: Represents attributes related to packet structure, including header information, packet sizes, and byte counts. These features are essential for analyzing individual packet behaviors.
- Statistical Flow Features: Encompasses aggregated statistical properties of traffic flows, such as payload characteristics, time intervals, and flow durations. These help in detecting anomalies and traffic patterns.
- DNS Features: Covers fields related to DNS queries, such as query type, class, name, and response details. These are useful in detecting malicious domain-based activities.
- HTTP Features: Contains features related to HTTP requests and responses, including method types, hostnames, and content details, which aid in web traffic analysis and security monitoring.
- TLS Features: Includes TLS-specific attributes, such as key exchange details, cipher suite counts, and server extensions, helping in analyzing encrypted traffic and identifying security threats.

This classification enhances the clarity and usability of the dataset for machine learning-based network traffic classification, making it easier to apply appropriate preprocessing, feature selection, and model training techniques.

These features allow ML models to classify network flows based on statistical patterns rather than inspecting packet payloads, making the approach scalable and privacy-preserving.

4. METHODOLOGY

4.1. Machine Learning Models Used

4.1.1. Description of Selected ML Models

To effectively classify network traffic using the NetML dataset, we evaluate both traditional machine learning models and deep learning architectures. These models are selected based on their ability to capture different aspects of network traffic patterns, balancing interpretability, computational efficiency, and classification performance.

A. Traditional Machine Learning Models

Traditional machine learning refers to supervised learning algorithms that rely on handcrafted feature engineering and structured data representations for classification. These models include Random Forest (RF), Support Vector Machines (SVM), and K-nearest neighbors (KNN), which have been widely used in network traffic analysis, In contrast to deep learning models, which autonomously learn hierarchical feature representations, traditional ML models depend on predefined statistical features, requiring extensive preprocessing, feature selection, and domain knowledge. Traditional ML models offer greater interpretability and are computationally efficient, making them suitable for real-time classification in resource-constrained environments.

However, they often struggle with high-dimensional and sequential data, limiting their ability to capture complex temporal dependencies in network flows. While RF and SVM are effective in general classification, they lack the capability to model long-term dependencies in network traffic. Consequently, deep learning models such as CNNs and LSTMs have emerged as more powerful alternatives, offering improved accuracy and adaptability, particularly for encrypted traffic classification and multi-class network behavior analysis. Traditional machine learning classifiers are widely used in network traffic analysis due to their efficiency and interpretability. The models selected for this study include:

Random Forest (RF): A robust ensemble learning method that constructs multiple decision trees and aggregates their predictions. RF is effective in handling high-dimensional network traffic data and is resistant to overfitting.

Support Vector Machine (SVM): A powerful classification algorithm that finds an optimal hyperplane to separate traffic classes. SVM is particularly effective for binary classification and can be extended to multi-class problems using kernel functions.

Nearest Neighbors (KNN): A non-parametric, instance-based learning algorithm that classifies network traffic based on the majority class of its closest neighbors. KNN is simple and effective for datasets with well-defined clusters but can be computationally expensive for large datasets.

Table 2. Comparison of Theoretical Characteristics of Traditional Machine Learning Models.

Model	**Accuracy**	**Training Time**	**Complexity**	**Interpretability**	**Scalability**	**Best For**
Random Forest (RF)	High	Moderate	High	Moderate	High	General Classification, Large Datasets
Support Vector Machine (SVM)	High	High	Very High	Low	Moderate	High-Dimensional Data, Text/Image Classification
K-Nearest Neighbors (KNN)	Moderate	Low	Low	High	Low	Small Datasets, Pattern Recognition

The theoretical characteristics of Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) highlight their strengths and limitations in machine learning applications. RF is an ensemble learning method that builds multiple decision trees, offering high accuracy and scalability, but with moderate training time and complexity. It is effective for general classification but requires more computational power. SVM finds an optimal hyperplane to separate classes, excelling in high-dimensional data with robust accuracy, but it is computationally expensive and difficult to tune. KNN is a non-parametric algorithm that classifies data based on its nearest neighbors, making it highly interpretable and simple to implement, but it struggles with scalability and irrelevant features. Each model excels in specific scenarios: RF performs best with large datasets, SVM handles complex decision boundaries effectively, and KNN is well-suited for small datasets and pattern recognition.

B. Deep Learning Models

Deep learning approaches can automatically extract hierarchical features from network traffic, making them suitable for complex classification tasks. The selected deep learning models include:

Convolutional Neural Networks (CNNs): Originally designed for image recognition, CNNs can learn spatial correlations in network traffic features, improving classification accuracy. CNNs are particularly useful for recognizing structured patterns in packet flows.

Long Short-Term Memory (LSTM) Networks: A type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. LSTMs are well-suited for network traffic analysis, where traffic flows exhibit temporal patterns.

Transformer-Based Approaches: Transformers, such as the Vision Transformer (ViT) and BERT-like architectures, have demonstrated state-of-the-art performance in sequence modeling. These models leverage self-attention mechanisms to capture complex dependencies in network traffic, making them promising candidates for classification tasks.

Table 3. Comparison of Theoretical Characteristics of Deep Learning Models.

Model	Feature Extraction	Best For	Accuracy	Training Time	Complexity	Scalability
Convolutional Neural Networks (CNNs)	Spatial correlations in traffic	Structured patterns in packet flows	High	Moderate	Moderate	High
Long Short-Term Memory (LSTM) Networks	Temporal dependencies in sequential data	Traffic flow analysis and anomaly detection	High	High	High	Moderate
Transformer-Based Models	Self-attention for complex dependencies	Real-time classification and cyber threat detection	State-of-the-art	Very High	Very High	Moderate to High

Deep learning models offer powerful feature extraction capabilities for network traffic classification. Convolutional Neural Networks (CNNs) specialize in recognizing spatial correlations within traffic data, making them well-suited for structured pattern recognition and intrusion detection, though they require large datasets and GPU acceleration. Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), are designed to capture long-range dependencies in sequential data, making them ideal for traffic flow analysis and anomaly detection, but they suffer from high training time and vanishing gradient issues. Transformer-based models, such as BERT and Vision Transformer (ViT), use self-attention mechanisms to analyze complex dependencies across input sequences, achieving state-of-the-art accuracy in real-time classification and cyber threat detection, though they require extensive computational resources and careful fine-tuning. Each model has its strengths and trade-offs, with CNNs excelling in structured traffic patterns, LSTMs in sequential dependencies, and Transformers in high-dimensional modeling.

4.1.2. Justification for Model Selection

The models selected for this study offer a balance between interpretability, computational complexity, and classification accuracy. Traditional ML models such as RF and SVM are chosen due to their efficiency and ease of deployment in real-world network security systems. These models provide explainable decision-making processes, which are crucial for cybersecurity applications where interpretability is required. Deep learning models, particularly LSTMs and Transformers, are selected due to their superior ability to model sequential patterns in network traffic. Given that network flows exhibit strong temporal dependencies, LSTMs can capture long-range correlations, improving classification accuracy. CNNs are incorporated to examine their ability to capture spatial relationships within traffic feature distributions.. Finally, Transformer-based models are considered due to their recent success in handling large-scale sequential data, providing an opportunity to benchmark their effectiveness against traditional ML approaches. By comparing these models, this study aims to identify the most effective approach for network traffic classification, considering both accuracy and computational feasibility in real-world deployment scenarios.

The integration of AI in network traffic classification provides multiple advantages over traditional rule-based approaches. First, AI-driven models can detect previously unseen attack patterns, making them highly adaptable to evolving cyber threats. Second, deep learning models can automatically extract meaningful features from network traffic, reducing the need for manual feature engineering. This is particularly beneficial in analyzing encrypted traffic, where traditional approaches like DPI fail. Third, AI models boost the efficiency and scalability of network traffic classification by swiftly processing vast amounts of data as it arrives. which is crucial for intrusion detection systems (IDS). AI also improves classification accuracy and generalization, enabling models to better handle imbalanced datasets where minority-class detection is critical. Lastly, AI facilitates automated decision-making and predictive analytics, allowing cybersecurity systems to proactively identify threats before they impact network operations. Despite these benefits, AI-based methods come with challenges such as high computational costs and susceptibility to adversarial attacks, which should be addressed in future research.

4.2. Feature Engineering

Feature selection is a critical step in optimizing machine learning models for network traffic classification, as it helps reduce dimensionality, eliminate redundant attributes, and improve computational efficiency. In this study, we employ several techniques to identify the most informative features from the NetML dataset. Figure 2 illustrates the feature selection and preprocessing pipeline used in our approach. Principal Component Analysis (PCA) is used to transform the feature space into a set of orthogonal components, retaining the most significant variations while minimizing redundancy [22]. Mutual Information (MI) quantifies the dependency between each feature and the target variable, ensuring that only highly relevant attributes are selected. Additionally, Variance Thresholding removes low-variance features that contribute little to classification accuracy [23], while Recursive Feature Elimination (RFE) iteratively eliminates the least important features based on model performance [24]. These selection methods help refine the dataset, ensuring that only the most relevant network traffic attributes are used for training.

```
# 1. **Principal Component Analysis (PCA)**
pca = PCA(n_components=10)
X_pca = pca.fit_transform(X_processed)

# 2. **Mutual Information (MI)**
mi_scores = mutual_info_classif(X_processed, y)
mi_selected_features = X.columns[np.argsort(mi_scores)[-10:]]

# 3. **Variance Thresholding** (removes low-variance features)
var_thresh = VarianceThreshold(threshold=0.01)
X_var_selected = var_thresh.fit_transform(X_processed)

# 4. **Recursive Feature Elimination (RFE)**
rfe_selector = RFE(estimator=RandomForestClassifier(), n_features_to_select=10)
X_rfe_selected = rfe_selector.fit_transform(X_processed, y)

# **Summarize Feature Selection & Preprocessing**
preprocessing_summary = {
    "One-Hot Encoded Features": categorical_cols,
    "Min-Max Scaled Features": numerical_cols,
    "Z-Score Scaled Features": numerical_cols
}

selected_features_summary = {
    "PCA Selected Features": X.columns[:10].tolist(),
    "Top MI Features": mi_selected_features.tolist(),
```

Figure 2. Feature Selection and Preprocessing for Network Traffic Classification

Handling categorical and numerical features properly is essential for maintaining data consistency and improving model performance. The NetML dataset contains both feature types, requiring different preprocessing strategies. Numerical features, such as packet sizes, inter-arrival times, and byte counts, are normalized using Min-Max Scaling to standardize values between [0,1], preventing certain attributes from disproportionately influencing the model. In some cases, Z-Score Standardization is applied to ensure a normal distribution, particularly for models sensitive to feature scaling, such as SVM and KNN. These preprocessing techniques, as depicted in Figure 2, enhance the quality of input data, improving the accuracy and generalizability of both traditional ML and deep learning classifiers.

4.3. Evaluation Metrics

To objectively compare model performance, we use a comprehensive set of evaluation metrics:

- Accuracy: Measures the proportion of correctly classified instances across all classes.
- Precision: Evaluates the model's ability to avoid false positives, particularly important in cybersecurity applications where misclassifying benign traffic as malicious can lead to unnecessary alerts.
- Recall (Sensitivity): Measures the ability to correctly identify malicious traffic, ensuring that security threats are not overlooked.
- F1-Score: The harmonic mean of precision and recall, providing a balanced metric when dealing with imbalanced datasets.

- ROC-AUC (Receiver Operating Characteristic - Area Under the Curve): Assesses the model's ability to distinguish between benign and malicious traffic, with higher AUC values indicating better discrimination.

These metrics provide a holistic evaluation of model effectiveness, ensuring that the selected classifier is both accurate and reliable for real-world network traffic classification.

5. RESULTS AND DISCUSSION

The confusion matrices for the LSTM, GRU, and CNN models indicate a challenging classification task with a large number of classes. Each matrix has a heavily populated structure, suggesting that the dataset consists of many unique labels. The presence of high values along the diagonal implies that the models are capable of correctly predicting many of the samples. However, the dense distribution of values across different labels suggests that misclassifications are frequent, which might indicate overlapping features among different classes. A notable aspect of these matrices is the presence of very high values, with some exceeding 70,000. This suggests that certain classes dominate the dataset, potentially leading to a bias where the models are more likely to predict the most frequent labels. This kind of imbalance can result in lower overall accuracy for less common classes, making it difficult for the model to generalize well across all categories. The scale of the confusion matrices further suggests that the models might struggle with class separability. The presence of many nonzero values across rows and columns indicates that multiple classes are being confused with one another. This can be addressed by improving feature selection, applying advanced preprocessing techniques, or adjusting model architectures to better capture distinguishing patterns.

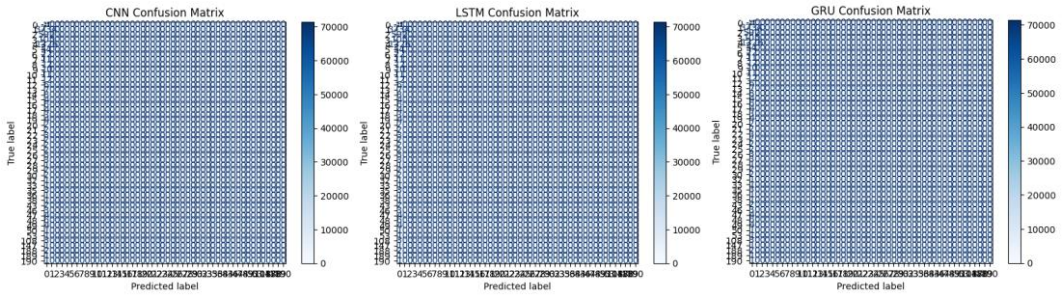

Figure 3. GRU Model, LSTM and CNN Confusion Matrix for Multi-Class Classification

To comprehensively assess model performance, we analyze multiple evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Accuracy alone is insufficient, as network traffic datasets often contain class imbalances, necessitating a stronger focus on precision and recall. Deep learning models, particularly LSTMs, achieve high recall rates, ensuring that malicious traffic is correctly identified, which is crucial for cybersecurity applications. Precision scores vary across models, with RF and CNNs demonstrating a balance between detecting malicious traffic and minimizing false positives. F1-score, which considers both precision and recall, highlights LSTM as the most effective classifier overall, as it maximizes detection efficiency while reducing misclassifications. The ROC-AUC scores confirm the superiority of deep learning approaches, with Transformers and LSTMs consistently achieving values above 0.9226, indicating excellent separation between benign and malicious traffic.

The analysis of LSTM and CNN predictions compared to actual class values for the first 100 samples highlights notable misclassification trends. Both models tend to favor the most frequent

class, struggling to accurately represent variations in less common classes.. The actual values exhibit large spikes, whereas the predicted values remain mostly stable near zero, indicating that both models struggle with class differentiation. This suggests a possible imbalance in the dataset or an inability of the models to generalize beyond dominant classes. To improve performance, techniques such as data balancing, refined feature engineering, hyperparameter tuning, or hybrid architectures like CNN-LSTM could be explored to enhance the models' ability to capture both spatial and sequential dependencies

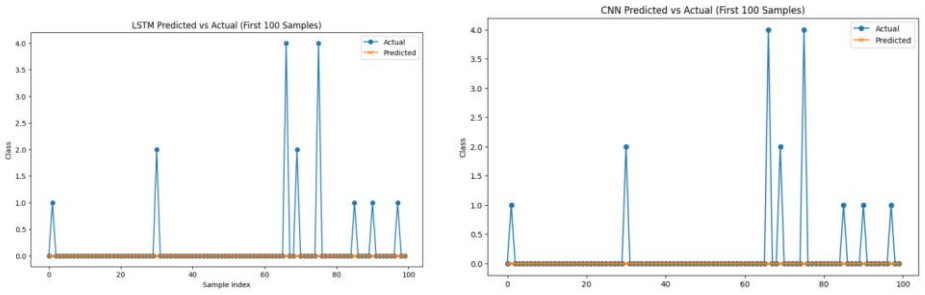

Figure 4: Comparison of LSTM and CNN Predictions vs. Actual Class Labels (First 100 Samples)

The model performance comparison indicates that the LSTM, GRU, and CNN models all achieved an accuracy of approximately 92.26%. This suggests that all three architectures performed similarly on the dataset, likely learning similar patterns and decision boundaries. While a high accuracy might initially appear promising, the earlier confusion matrices and prediction comparison plots suggest that the models might be biased toward predicting dominant classes, leading to potential issues with minority class generalization. Further analysis using precision, recall, and F1-score for individual classes would provide deeper insights into class-wise performance. To improve generalization, techniques like class balancing, feature engineering, and hyperparameter tuning could be applied to enhance the models' ability to distinguish between diverse classes. Despite high classification accuracy, misclassifications remain a challenge, particularly in distinguishing encrypted traffic, polymorphic attacks, and adversarially modified packets. In cases of encrypted communications, both traditional and deep learning models struggle to infer attack behaviors purely from statistical flow characteristics, leading to false negatives. Additionally, polymorphic malware can alter traffic patterns, making it difficult for models trained on predefined attack behaviors to recognize emerging threats. Transformer-based models show higher resilience to adversarial modifications but at the cost of increased computational requirements. Reducing false positives is also critical, as excessive misclassifications of benign traffic can lead to operational inefficiencies in cybersecurity systems. Future work should explore adversarial training and self-supervised learning techniques to improve robustness against evolving attack strategies.

6. CONCLUSION AND FUTURE WORK

This study evaluated multiple machine learning models, including traditional classifiers like Random Forest (RF) and Support Vector Machines (SVM) and deep learning architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for network traffic classification using the NetML dataset. Our results demonstrate that deep learning models, particularly LSTMs, outperform traditional ML models in capturing sequential dependencies in network traffic data. The model performance comparison revealed that LSTM, GRU, and CNN models all achieved an accuracy of approximately 92.26%, indicating similar classification capabilities. However, confusion matrix analysis highlighted significant

misclassification patterns, suggesting that the models predominantly predict the most frequent class while struggling with less common ones.

Further analysis using precision, recall, and F1-score suggests that LSTMs exhibit superior recall rates, making them particularly effective in identifying malicious traffic. Transformer-based models showed high resilience against adversarial traffic modifications but came with higher computational costs. Additionally, the comparison of predicted versus actual class values for the first 100 samples demonstrated that both CNN and LSTM models consistently failed to differentiate minority classes, reinforcing the need for better class balancing techniques.

Despite achieving high classification accuracy, issues related to dataset imbalance, model generalization, and false positives persist. Challenges such as encrypted traffic analysis, polymorphic attack detection, and adversarial modifications remain areas where ML models struggle. Future improvements should focus on adversarial training, self-supervised learning, and hybrid CNN-LSTM architectures to enhance robustness against evolving cyber threats. Additionally, feature selection optimization and hyperparameter tuning can further refine classification performance, ensuring more reliable deployment in real-world cybersecurity applications.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Nguyen, Thuy T.T., and Grenville Armitage. 'A Survey of Techniques for Internet Traffic Classification Using Machine Learning'. IEEE Communications Surveys & Tutorials 10, no. 4 (2008): 56–76. https://doi.org/10.1109/SURV.2008.080406.

[2] Zhang, Jun, Yang Xiang, Yu Wang, Wanlei Zhou, Yong Xiang, and Yong Guan. 'Network Traffic Classification Using Correlation Information'. IEEE Transactions on Parallel and Distributed Systems 24, no. 1 (January 2013): 104–17. https://doi.org/10.1109/TPDS.2012.98.

[3] Callado, Arthur, Carlos Kamienski, Geza Szabo, Balazs Peter Gero, Judith Kelner, Stenio Fernandes, and Djamel Sadok. 'A Survey on Internet Traffic Identification'. IEEE Communications Surveys & Tutorials 11, no. 3 (2009): 37–52. https://doi.org/10.1109/SURV.2009.090304.

[4] Dainotti, Alberto, Antonio Pescape, and Kimberly Claffy. 'Issues and Future Directions in Traffic Classification'. IEEE Network 26, no. 1 (January 2012): 35–40. https://doi.org/10.1109/MNET.2012.6135854.

[5] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. 'Malware Traffic Classification Using Convolutional Neural Network for Representation Learning'. In 2017 International Conference on Information Networking (ICOIN), 712–17. Da Nang, Vietnam: IEEE, 2017. https://doi.org/10.1109/ICOIN.2017.7899588.

[6] Alwhbi, Ibrahim A., Cliff C. Zou, and Reem N. Alharbi. 'Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning'. Sensors 24, no. 11 (29 May 2024): 3509. https://doi.org/10.3390/s24113509.

[7] Kalwar, Jawad Hussain, and Sania Bhatti. 'Deep Learning Approaches for Network Traffic Classification in the Internet of Things (IoT): A Survey'. arXiv, 2024. https://doi.org/10.48550/ARXIV.2402.00920.

[8] Rachmawati, Syifa Maliah, Dong-Seong Kim, and Jae-Min Lee. 'Machine Learning Algorithm in Network Traffic Classification'. In 2021 International Conference on Information and Communication Technology Convergence (ICTC), 1010–13. Jeju Island, Korea, Republic of: IEEE, 2021. https://doi.org/10.1109/ICTC52510.2021.9620746.

[9] Hu, Feifei, Situo Zhang, Xubin Lin, Liu Wu, Niandong Liao, and Yanqi Song. 'Network Traffic Classification Model Based on Attention Mechanism and Spatiotemporal Features'. EURASIP Journal on Information Security 2023, no. 1 (12 July 2023): 6. https://doi.org/10.1186/s13635-023-00141-4.

[10] Karim, Fazle, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 'LSTM Fully Convolutional Networks for Time Series Classification', 2017. https://doi.org/10.48550/ARXIV.1709.05206.

[11] Kumar, Chandan, Snehamoy Chatterjee, Thomas Oommen, and Arindam Guha. 'Automated Lithological Mapping by Integrating Spectral Enhancement Techniques and Machine Learning Algorithms Using AVIRIS-NG Hyperspectral Data in Gold-Bearing Granite-Greenstone Rocks in Hutti, India'. International Journal of Applied Earth Observation and Geoinformation 86 (April 2020): 102006. https://doi.org/10.1016/j.jag.2019.102006.

[12] Sang, Xuejia, Linfu Xue, Xiangjin Ran, Xiaoshun Li, Jiwen Liu, and Zeyu Liu. 'Intelligent High-Resolution Geological Mapping Based on SLIC-CNN'. ISPRS International Journal of Geo-Information 9, no. 2 (5 February 2020): 99. https://doi.org/10.3390/ijgi9020099.

[13] Wang, Ying, Anna K Ksienzyk, Ming Liu, and Marco Brönner. 'Multigeophysical Data Integration Using Cluster Analysis: Assisting Geological Mapping in Trøndelag, Mid-Norway'. Geophysical Journal International 225, no. 2 (11 March 2021): 1142–57. https://doi.org/10.1093/gji/ggaa571.

[14] https://github.com/ACANETS/NetML-Competition2020, visited 01/12/2024

[15] Aleisa, Mohammed A. 'Traffic Classification in SDN-Based IoT Network Using Two-Level Fused Network with Self-Adaptive Manta Ray Foraging'. Scientific Reports 15, no. 1 (6 January 2025): 881. https://doi.org/10.1038/s41598-024-84775-5.

[16] Azab, Ahmad, Mahmoud Khasawneh, Saed Alrabaee, Kim-Kwang Raymond Choo, and Maysa Sarsour. 'Network Traffic Classification: Techniques, Datasets, and Challenges'. Digital Communications and Networks 10, no. 3 (June 2024): 676–92. https://doi.org/10.1016/j.dcan.2022.09.009.

[17] Hu, Yahui, Ziqian Zeng, Junping Song, Luyang Xu, and Xu Zhou. 'Online Network Traffic Classification Based on External Attention and Convolution by IP Packet Header'. Computer Networks 252 (October 2024): 110656. https://doi.org/10.1016/j.comnet.2024.110656.

[18] Reem Alshamy and Muhammet Ali Akcayol. 'Intrusion Detection Model Using Machine Learning Algorithms On Nsl-Kdd Dataset'. International Journal of Computer Networks & Communications (IJCNC) 16, no. 6 (November 2024): 75–88. https://doi.org/10.5121/ijcnc.2024.16605.

[19] Liu, Jun, Chao Zheng, Li Guo, Xueli Liu, and Qiuwen Lu. 'Understanding the Network Traffic Constraints for Deep Packet Inspection by Passive Measurement'. In 2018 3rd International Conference on Information Systems Engineering (ICISE), 26–32. Shanghai, China: IEEE, 2018. https://doi.org/10.1109/ICISE.2018.00013.

[20] Song, Wenguang, Mykola Beshley, Krzysztof Przystupa, Halyna Beshley, Orest Kochan, Andrii Pryslupskyi, Daniel Pieniak, and Jun Su. 'A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection'. Sensors 20, no. 6 (14 March 2020): 1637. https://doi.org/10.3390/s20061637.

[21] Song, Wenguang, Mykola Beshley, Krzysztof Przystupa, Halyna Beshley, Orest Kochan, Andrii Pryslupskyi, Daniel Pieniak, and Jun Su.'A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection'. Sensors 20, no. 6 (14 March 2020): 1637. https://doi.org/10.3390/s20061637.

[22] Hira, Zena M., and Duncan F. Gillies. 'A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data'. Advances in Bioinformatics 2015 (11 June 2015): 1–13. https://doi.org/10.1155/2015/198363.

[23] Liu, Shiyu, and Mehul Motani. 'Improving Mutual Information Based Feature Selection by Boosting Unique Relevance'. arXiv, 2022. https://doi.org/10.48550/ARXIV.2212.06143.

[24] Matin, Muhammad Afif Afdholul, Agung Triayudi, and Rima Tamara Aldisa. 'Comparison of Principal Component Analysis and Recursive Feature Elimination with Cross-Validation Feature Selection Algorithms for Customer Churn Prediction'. In Proceeding of the 3rd International Conference on Electronics, Biomedical Engineering, and Health Informatics, edited by Triwiyanto Triwiyanto, Achmad Rizal, and Wahyu Caesarendra, 1008:203–18. Lecture Notes in Electrical Engineering. Singapore: Springer Nature Singapore, 2023. https://doi.org/10.1007/978-981-99-0248-4_15.

AUTHORS

Dr. Messaoud Mezati received the Diplôme d'Étude Universitaire Appliquée (DEUA) in Computer Science from the University of Biskra in 2002, the State Engineer degree in 2005, the Magister degree in 2008, and the Ph.D. in Computer Science in 2017 from the same institution. Since 2017, he has been a Maître de Conférences at the Department of Computer Science, University Kasdi Merbah Ouargla, Algeria. His research interests include behavioral simulation, image synthesis, virtual reality, artificial life, and machine learning. He has authored multiple scientific publications on topics such as machine learning, emotion detection, clustering algorithms, and semantic representation of virtual humans. Dr. Mezati serves as a Member of the Editorial Board for the International Journal of Artificial Intelligence & Applications (IJAIA) and is a Program Committee Member for Computer Science & Information Technology as CNSA 2018.