A Machine Learning-Based Packet Sniffer for Detection and Classification of the Denial-Of-Service Attack Packets at the Network Layer

Kipkorir Peacemark*, Ephantus Mwangi and Jotham Wasike

Department of Pure and Applied Sciences, Kirinyaga University

***Correspondence Author**

DOI: https://doi.org/10.51584/IJRIAS.2025.100500051

Received: 09 May 2025; Accepted: 15 May 2025; Published: 09 June 2025

ABSTRACT

Cyber threats attacks have continued to evolve in complexity and sophistication, posing significant risks to an organization's network infrastructure and sensitive data's availability, confidentiality, and integrity. Therefore, there is a great need to create a defense mechanism to counteract this problem. This study therefore was focused on modeling a packet sniffer utilizing machine learning techniques to identify denial of service (DOS) attack packets at the network layer of the OSI model. The overall purpose of the study was to capture and interpret packets transmitted over a local area network to detect and capture the DOS threats within the Open Systems Interconnection Model (OSI) network layer. This layer is prone to several attacks for instance, denial-of-service, routing protocol attacks, Port scanning and enumeration, and fragmentation-based attacks. This study, delved into detecting and capturing the denial of service threats at the third layer of the OSI model in a local area network. Some examples of DOS attacks are UDP flood which sends a significant quantity UDP (User Datagram Protocol) packets to the targeted systems and thereby exhausting network resources, ICMP flood which transmits a significant quantity of Internet Control Message Protocol (ICMP) packets to overwhelm network devices, SYN flood which takes advantage of the TCP three-way hand-shake procedure by sending a lot of SYN requests without carrying out the necessary handshake, using server resources and blocking valid connections. Essential components extracted from Ethernet frames comprise TCP segments, ICMP packets, IPv4 packets, and associated flags. Although antivirus programs, intrusion detection systems, and firewalls are crucial barriers against malicious attacks, they frequently fall short in detecting and halting more crafty attacks that evade their protection. The study sought to bridge this gap by providing an automated machine learning-based packet sniffer that can identify and categorize network risks. The LightGBM model was successfully trained and implemented for the task of detecting DoS attacks. CICIDS2018 dataset was used, which provided labeled network traffic data containing both normal and attack (DoS) instances. The model was trained to classify traffic as either normal or a DoS attack based on various network features. The model's performance was evaluated using several metrics to demonstrate its ability to accurately detect threats at the network layer in a local area network including sensitivity, specificity, and accuracy. The AUC (Area Under the Curve) was particularly high, which indicated that the model was able to effectively differentiate between normal traffic and DoS attacks. Additionally, the F1-score, precision, and recall were balanced, suggesting that the model was capable of identifying attacks while minimizing false positives and false negatives. The model was successful in meeting its primary objective of detecting DoS attacks from network traffic. The performance metrics indicated that LightGBM is a strong candidate for the task, achieving a high AUC and a well-balanced F1-score. This showed that the model achieved good generalization capabilities, and it can effectively distinguish between normal traffic and DoS attack traffic in most cases. The main contribution of this work is the development of a LightGBM-based machine learning model for detecting DoS attacks using the CICIDS2018 dataset. The model's ability to classify network traffic as normal or malicious will aid in enhancing network security by automating the detection of such attacks in LANs. The model will henceforth serve as a foundational step for building more advanced intrusion detection systems, especially for environments where DoS attacks are prevalent.

Key words: DOS, Light GBM, LANs, Packet Sniffer, Network layer.

INTRODUCTION

Network security is now a top priority for businesses of all sizes in today's interconnected society. As networks continue to expand and become more complex, the potential for security breaches and cyber threats increases (Julian *et al*., 2014). To effectively protect network infrastructure and sensitive information, organizations must employ advanced techniques and technologies that go beyond traditional security measures. One such approach is the use of advanced packet sniffer analysis (Tariq *et al.*, 2023). Packet sniffers are tools that capture and analyze network traffic, allowing organizations to monitor and examine the data packets flowing through their networks. These tools provide valuable insights into network behavior, identifying potential vulnerabilities, anomalies, and malicious activities (Chiradeep, 2022). By analyzing packet-level information, Companies can obtain an extensive comprehension of the security posture of their network and make informed decisions to enhance network security.

Traditional packet sniffers however, fall short in addressing the evolving and sophisticated nature of cyber threats often. Firstly, in response to the widespread denial-of-service assaults conventional security techniques include intrusion detection/prevention systems (IDS/IPS) and firewalls may find it difficult to scale successfully. These assaults have the potential to overload network resources, making it impossible for these devices to instantly inspect and filter traffic. The second problem with the traditional systems is Resource Exhaustion. Denial-of-service (DoS) attacks frequently seek to deplete network resources, including memory, CPU, bandwidth, and connection state tables. The volume of malicious traffic may overwhelm traditional security systems, resulting in resource depletion, network performance deterioration, or total service failure. Detection Challenges is the third issue with the conventional systems. Real-time DoS attack detection can be difficult for conventional security solutions. It can be challenging for standard detection algorithms to discern between malicious and legitimate traffic since attackers may use a variety of evasion techniques to conceal their operations or disperse attack traffic over several sources. Zero-Day Attacks were the fourth issue with traditional systems: to detect and lessen threats, traditional security solutions usually rely on rule-based rules or signature-based detection. Nevertheless, these defenses may fail to identify zero-day DoS assaults, which take use of undiscovered vulnerabilities or attack paths, leaving networks open to exploitation. As a result of the mentioned challenges, organizations have over time faced an increased risk of security breaches, and service disruptions, potentially leading to financial losses, reputational damage, and legal liabilities (Abuya, 2020).

To address the limitations of traditional network security measures and enhance network security, the proposed solution was to implement a machine learning-based packet sniffer for identifying and categorizing the denial-of-service attacks packets at the network layer as a proactive and effective security measure. Through the application of cutting-edge technology such as machine learning, artificial intelligence, and behavioral analytics, organizations can acquire more profound understanding of network traffic, identify denial-of-service incidents, and promptly address possible security risks. Key components of the proposed solution included the designing, developing and deployment of an advanced packet sniffer tool that can capture and analyze network data in real time. This program analyzed trends, spot anomalies, and pinpointed possible security vulnerabilities in network traffic data using machine learning techniques. Furthermore, data collection and storage procedures were implemented to assure the preservation of comprehensive records for future study and investigation.

METHODOLOGY

Study Area

This study used experimental research methodology and the research targeted a representative organization that utilizes networks in their daily operations as a result of financial and time constraints. The study was to be subject to the availability, access and quality of the sampled institutional data. In the network layer of the Local area network, the study focused on the IPV4 and IPV6 address protocols. These addresses are fundamental for routing and addressing traffic at the OSI model's network layer. The focus of this research was centered on developing a model for detecting and classifying of the denial-of-service attacks at the OSI model's network layer in a local area network. Some examples of DOS attacks are UDP flood which sends a significant quantity of packets over User Datagram Protocol (UDP) to the target systems and thereby exhausting network resources, ICMP flood which Transmits a substantial quantity of Internet Control Message Protocol (ICMP) packets to overwhelm network devices, SYN flood which takes advantage of the TCP three-way hand-shake procedure by sending a lot of SYN requests without carrying out the necessary handshake, using server resources and blocking valid connections. The use of machine learning packet sniffers models for detection and classification of DOS attack packets at the network layer greatly improved network security posture, improved threat detection capabilities, and helped enterprises stay ahead of cyber threats in an increasingly complex and dynamic threat landscape. The model was implemented in Python and employed the CICIDS2018 dataset downloaded from Kaggle.

Dataset Description

The CICIDS2018 dataset designed by the Canadian Institute for Cybersecurity (CIC), contained network traffic data collected from various network environments, including LANs. They contain normal traffic and many forms of attacks, including those targeting Layer 3 protocols, making them ideal for training a network security model. The dataset was used to train, validate, and test the model to evaluate trends, detect anomalies, and identify potential security vulnerabilities in network traffic data. The purpose of employing the CICIDS2018 dataset was to use the dataset to develop a model with comprehension and categorization of network traffic data. The model's performance was evaluated using several measures that demonstrated its ability to accurately identify threats at the network layer in a local area network, such as sensitivity, specificity, and accuracy.

https://www.kaggle.com/datasets?search=CICIDS2018+dataset is the URL to the dataset.

Datatypes

While dealing with the CICIDS2018 dataset for threat detection in network traffic OSI model’s network layer, this study used the following types of data: -

i. **Numerical data** - includes continuous or discrete numerical values representing various features or attributes of network traffic data. The CICIDS2018 dataset included numerical data such as packet sizes, flow durations, byte counts, and numerical representations of protocol types, IP addresses for the source and destination, and port numbers.

ii. **Categorical data** - are discrete categories or labels for various attributes in a dataset. The CICIDS2018 dataset contained categorical data such as protocol names (e.g., TCP, UDP), attack types (e.g., DDoS, brute force), and binary labels denoting normal or malicious traffic.

iii. **Text Data** - includes data such as domain names, URLs, or payload content. Text data required particular preprocessing techniques such as tokenization, stemming, or vectorization before utilizing it as input characteristics for machine learning models.

iv. **Temporal data** – This refers to timestamps or time-related information connected with network traffic occurrences. This comprises timestamps showing the start and end times of network flows, as well as time intervals between packets.

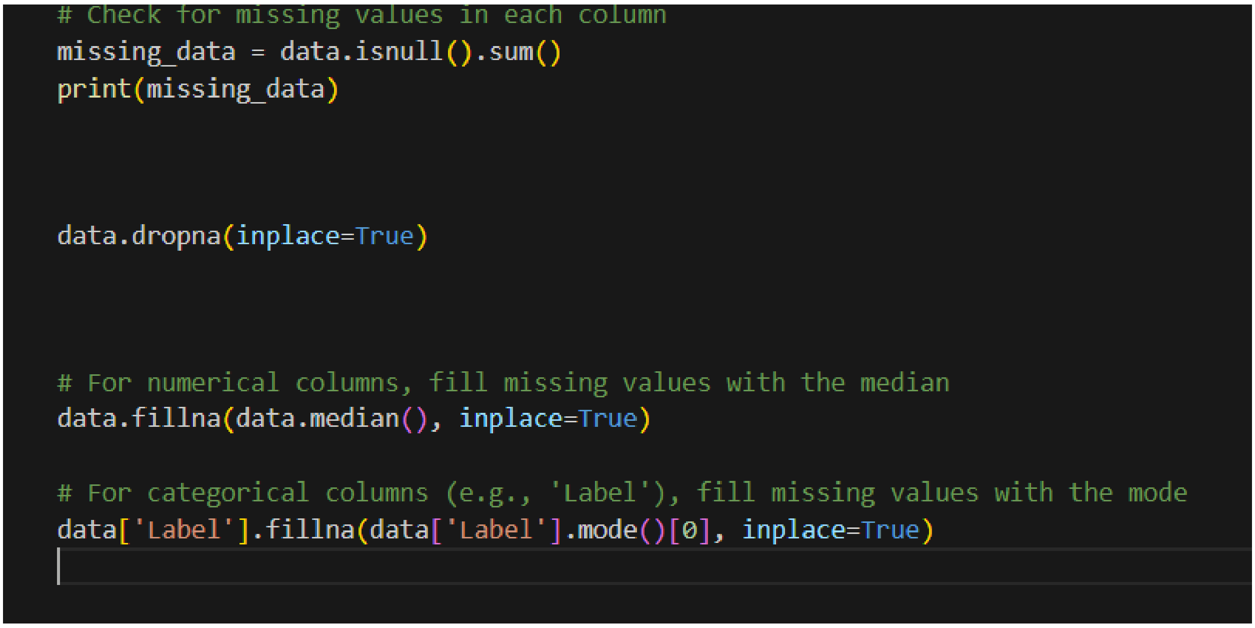
v. **Spatial data** - represented geographical information such as IP addresses or network locations.

vi. **Image Data** - included picture data created from visual representations of network traffic or network diagrams.

The **CICIDS 2018** dataset being a comprehensive collection of network traffic data and with well labeled attack types and normal traffic designed for cybersecurity research became ideal for this study. The dataset was publicly available, and could be accessed via the Canadian Institute for Cybersecurity website. The data was also downloaded in CSV format. The dataset contained both normal and attack traffic and the attacks included Denial of service and Distributed denial of service and all these data was formatted in CSV format. The attributes included flow features such as source and destination Ips and port numbers, protocol features and payload data of the network packets, time related features(timestamps), connection features (for instance, the number of connections from an IP address) and flags and state information (whether a connection is in an established state or closed).

Data Cleaning and Preprocessing

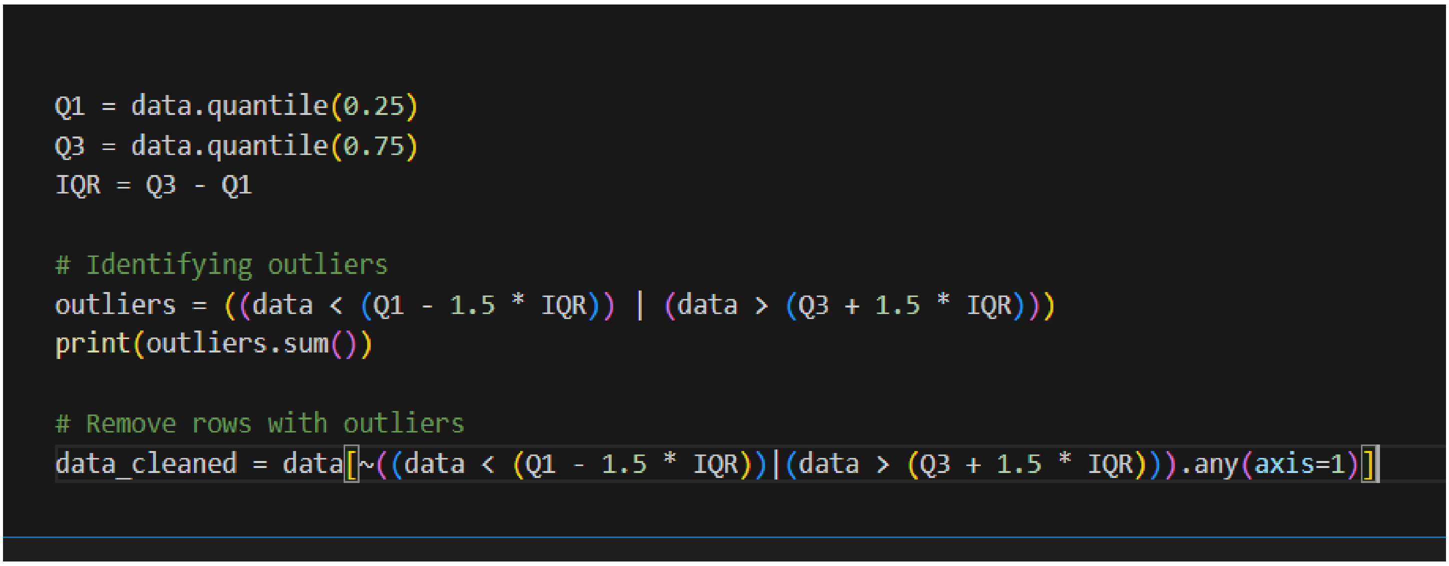
To ensure that the dataset was accurate, complete and formatted correctly, the dataset was first loaded and its contents inspected to understand the structure and identify any potential issue. Once the data was loaded, the following code snippet was employed to check for the missing data in the dataset. Having identified the missing values, both in rows and columns, all the missing rows were removed. Consequently, for the columns with numerical data, the missing values were filled with the mean or median depending on the distribution of the data and mode for the categorical data as shown in figure 1 below.



```
# Check for missing values in each column
missing_data = data.isnull().sum()
print(missing_data)



data.dropna(inplace=True)



# For numerical columns, fill missing values with the median
data.fillna(data.median(), inplace=True)

# For categorical columns (e.g., 'Label'), fill missing values with the mode
data['Label'].fillna(data['Label'].mode()[0], inplace=True)
```

Figure 1: Image showing the process by which the study used in removing missing values

To avoid bias and distorted results, duplicate rows and columns were also removed. Since outliers can skew results, they were identified using interquartile range (IQR) and removed as shown in figure 2 below.

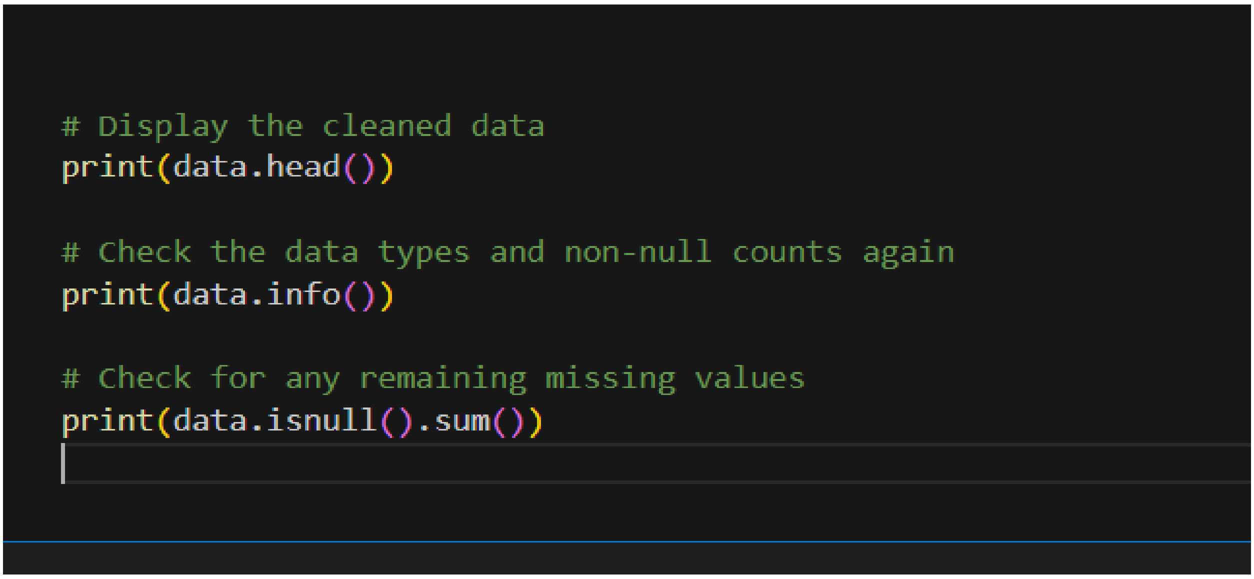


```
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1

# Identifying outliers
outliers = ((data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 * IQR)))
print(outliers.sum())

# Remove rows with outliers
data_cleaned = data[~((data < (Q1 - 1.5 * IQR))|(data > (Q3 + 1.5 * IQR))).any(axis=1)]
```

Figure 2: A screenshot showing outliers removal snippet

All categorical columns need to be encoded into numerical formats for machine learning models- label encoding for binary classification by converting them into binary values (0 and 1).

To confirm that our data is finally clean and ready for modelling the following code was used.



```
# Display the cleaned data
print(data.head())

# Check the data types and non-null counts again
print(data.info())

# Check for any remaining missing values
print(data.isnull().sum())
```

Figure 3: Clean data confirmation snippet

These preprocessing steps collectively prepared the CICIDS2018 dataset effectively facilitating the extraction of meaningful insights and the development of accurate machine learning based packet sniffer model for effective and efficient detection of DOS attacks at the network layer in a LAN.

Data Splitting

Data splitting was very fundamental and it was geared towards dividing the dataset into distinct subsets, each serving a specific purpose in model training, evaluation and validation. The preprocessed CICIDS2018 dataset went through division into two major subsets: the training set and test set in the ratio 80% to 20% respectively.

To perform this splitting the researcher employed the train_test_split () from sklearn.model selection. The dataset was split into two variables features (X) and target variable (Y). Feature (X) variables representing the input/independent variables and the Target (Y) variables representing the output variable that the model tries to predict. Target variable in our CICIDS2018 dataset is the label column showing the Normal/Attack traffic and the rest are feature columns.

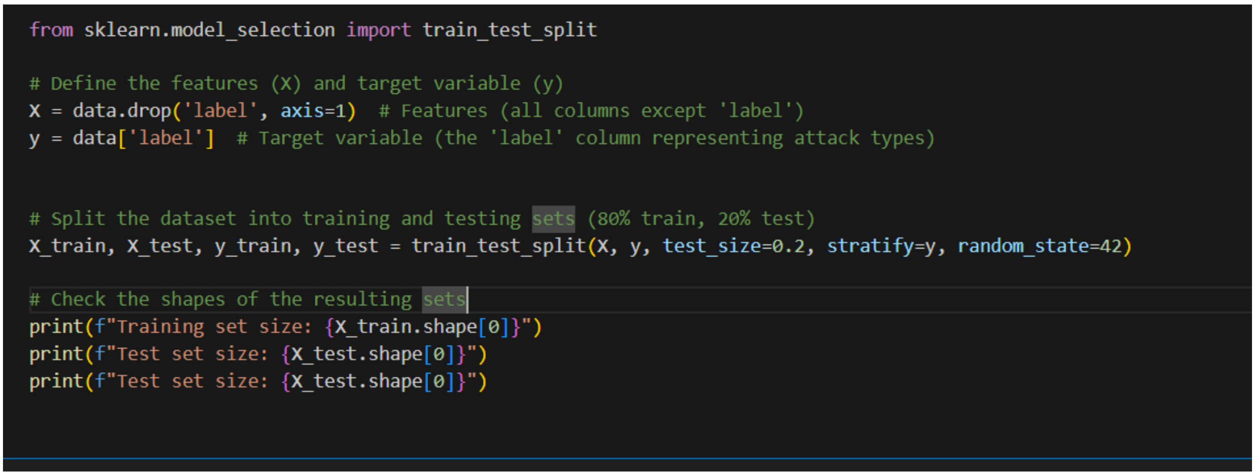
The training set formed the foundation for model training, enabling optimization of parameters and the learning of underlying patterns. It exposes the model to diverse examples, facilitating its ability to generalize to unseen instances.

The test set helps fine-tune hyperparameters and prevent overfitting by monitoring model performance during training. Regular evaluation on the validation set aids in adjusting parameters for optimal performance and generalization.

The validation set was created to serve as an unbiased benchmark to evaluate the final model's performance and assess its ability to generalize. It provides an objective measure of the model's accuracy and other metrics, offering insights into its real-world effectiveness. This third set was created by splitting the training data further.

The Stratify parameter was employed to ensure that the class distribution is similar in both the training and test sets.

The splitting of the dataset was done as shown in figure 4 below.



```
from sklearn.model_selection import train_test_split

# Define the features (X) and target variable (y)
X = data.drop('label', axis=1)  # Features (all columns except 'label')
y = data['label']  # Target variable (the 'label' column representing attack types)


# Split the dataset into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)

# Check the shapes of the resulting sets
print(f"Training set size: {X_train.shape[0]}")
print(f"Test set size: {X_test.shape[0]}")
print(f"Test set size: {X_test.shape[0]}")
```

Figure 4: dataset splitting code

Adherence to best practices in data splitting was key to prevent data leakage and ensured the reproducibility and integrity of the evaluation process.

Light Gradient Boosting Machines (LightGBM) Model

LightGBM (Light Gradient Boosting Machine) is a decision tree-based learning algorithm, highly efficient, distributed, and scalable implementation of gradient boosting that was used in this research.

The decision to opt for an LightGBM approach was informed by its inherent strengths in speed and accuracy, making it a great choice for classification tasks, including DoS (Denial of Service) attack detection and also using histogram-based algorithms for binning continuous features, reducing memory usage and speeding up training. Unlike traditional depth-wise tree growth, LightGBM employs a leaf-wise tree growth technique, which grows trees by splitting the leaf with the greatest loss reduction.

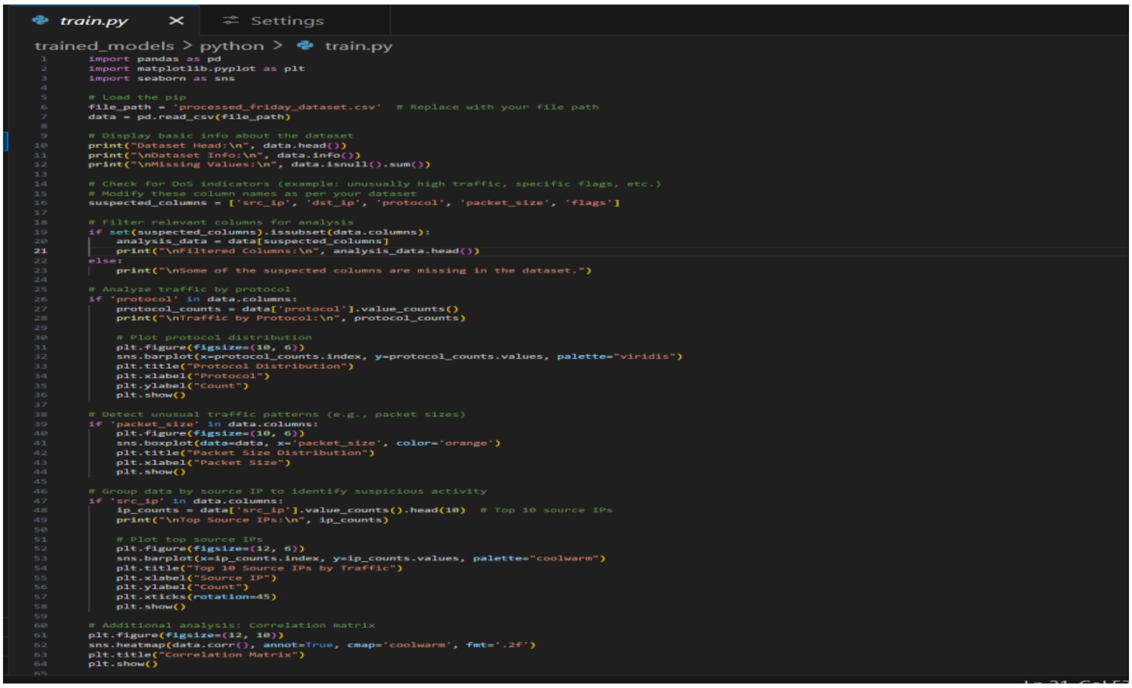
Furthermore, the empirical evidence supporting the good performance of LightGBM in that it outperforms traditional machine learning models like Random Forest and SVM in many real-world applications solidified its suitability for the development of the a machine learning packet sniffer model for the detection and classification of DOS attacks at the network layer of the OSI model. Numerous studies and benchmarking experiments have highlighted the superior performance of LightGBM in effectively handling imbalanced datasets, which is common in network attack detection tasks for instance, normal traffic is much more frequent than attack traffic.

Training

Training the model involved the iterative process of updating the model's parameters using the training dataset to minimize the binary cross-entropy loss, which served as the optimization objective. The training dataset, comprising a subset of the CICIDS2018 dataset, provided the foundational examples necessary for the model to learn meaningful representations of normal traffic and Denial of service attacks. Through the process of forward and backward propagation, the model iteratively adjusted its internal parameters to minimize the discrepancy between the predicted probabilities and the ground actual labels associated with each data in the training dataset.

The choice of binary cross-entropy loss as the optimization objective was driven by the binary nature of the prediction task, which involved discriminating between normal traffic and denial of service attacks. Binary cross-entropy loss, also known as log loss, quantifies the disparity between the model's predicted probabilities and the actual binary labels, penalizing deviations from the ground truth labels in a probabilistic manner. By minimizing the binary cross-entropy loss, the model learns to generate more accurate and calibrated predictions, thereby enhancing its discriminative power and predictive accuracy on unseen data.

The training process involved partitioning the training dataset into mini-batches, with each mini-batch containing a subset of examples randomly sampled from the training dataset. Throughout the training process the model updates its parameters based on the gradients computed on each mini-batch, enabling efficient utilization of computational resources and facilitating scalability to large datasets. By iteratively processing mini-batches and updating parameters, the model gradually refined its internal representations and learns to capture the underlying patterns and relationships present in the training data. The figure below shows code snippet for model training.



```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the pip
file_path = 'processed_friday_dataset.csv'  # Replace with your file path
data = pd.read_csv(file_path)

# Display basic info about the dataset
print("Dataset Head:\n", data.head())
print("\nDataset Info:\n", data.info())
print("\nMissing Values:\n", data.isnull().sum())

# Check for DoS indicators (example: unusually high traffic, specific flags, etc.)
# Modify these column names as per your dataset
suspected_columns = ['src_ip', 'dst_ip', 'protocol', 'packet_size', 'flags']

# Filter relevant columns for analysis
if set(suspected_columns).issubset(data.columns):
    analysis_data = data[suspected_columns]
    print("\nFiltered Columns:\n", analysis_data.head())
else:
    print("\nSome of the suspected columns are missing in the dataset.")

# Analyze traffic by protocol
if 'protocol' in data.columns:
    protocol_counts = data['protocol'].value_counts()
    print("\nTraffic by Protocol:\n", protocol_counts)

    # Plot protocol distribution
    plt.figure(figsize=(10, 6))
    sns.barplot(x=protocol_counts.index, y=protocol_counts.values, palette="viridis")
    plt.title("Protocol Distribution")
    plt.xlabel("Protocol")
    plt.ylabel("Count")
    plt.show()

# Detect unusual traffic patterns (e.g., packet sizes)
if 'packet_size' in data.columns:
    plt.figure(figsize=(10, 6))
    sns.boxplot(data=data, x='packet_size', color='orange')
    plt.title("Packet Size Distribution")
    plt.xlabel("Packet Size")
    plt.show()

# Group data by source IP to identify suspicious activity
if 'src_ip' in data.columns:
    ip_counts = data['src_ip'].value_counts().head(10)  # Top 10 source IPs
    print("\nTop Source IPs:\n", ip_counts)

    # Plot top source IPs
    plt.figure(figsize=(12, 6))
    sns.barplot(x=ip_counts.index, y=ip_counts.values, palette="coolwarm")
    plt.title("Top 10 Source IPs by Traffic")
    plt.xlabel("Source IP")
    plt.ylabel("Count")
    plt.xticks(rotation=45)
    plt.show()

# Additional analysis: Correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Matrix")
plt.show()
```

Figure 5: training code snippet

Hyperparameters Tuning

Hyperparameters tuning represents a crucial phase in the development of the machine Learning packet sniffer model for detection and classification of DOS at the network layer of the OSI model, as it involves optimizing various model parameters to enhance its performance and generalization ability. During the development phase, several key hyperparameters were subjected to fine-tuning; including the learning rate, batch size and dropout rates. Hyperparameters tuning was conducted using the evaluation dataset, which served as an independent benchmark for evaluating the model's performance across different hyperparameters configurations and selecting the optimal settings.

The learning rate, a fundamental hyperparameters in machine learning optimization algorithms, controls the magnitude of parameter updates during training and influences the convergence speed and stability of the training process. By adjusting the learning rate, this study managed to strike a balance between rapid convergence and overshooting, ensuring that the model effectively learns the underlying patterns present in the training data without being stuck in local minima or oscillating around the optimal solution. Through systematic experimentation and validation on the validation dataset, the optimal learning rate was identified to facilitate efficient training dynamics and robust model performance.

Another critical hyperparameters subject to tuning was the batch size, which determined the number of examples processed in each iteration of the training process. Batch size optimization involved balancing computational efficiency, memory constraints, and statistical efficiency to ensure effective parameter updates and convergence to the global optimum. By systematically varying the batch size and monitoring the model's performance on the validation dataset, we identified the optimal batch size that maximized training efficiency while maintaining stable and reliable learning dynamics.

Additionally, dropout rates, which controlled the fraction of neurons deactivated during training, were fine-tuned to prevent overfitting and improve model generalization. Dropout regularization encouraged the model to learn more robust and generalized representations by introducing stochasticity and diversity into the training process. By systematically varying dropout rates and evaluating the model's performance on the validation dataset, this study identified the optimal dropout rates that minimized overfitting while preserving the model's capacity to capture relevant patterns in the CICIDS2018 dataset.

The hyperparameters tuning played a pivotal role in optimizing the machine Learning packet sniffer model's performance and generalization ability. By fine-tuning key hyperparameters such as learning rate, batch size and dropout rates, we ensured that the model effectively learns and generalizes from the training data while minimizing overfitting and improving predictive accuracy.

Model Evaluation

The evaluation of the LightGBM model represented a critical phase in assessing its efficiency and reliability in real-world applications. Leveraging the independent testing dataset, we conducted a comprehensive analysis of the model's performance, employing a suite of relevant evaluation metrics to gauge its effectiveness in detecting and classifying the DOS attacks at the network layer of the OSI model.

The evaluation process encompassed the computation of a range of evaluation metrics, including accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics provided comprehensive insights into different aspects of the model's performance, enabling an understanding of its strengths and limitations in detecting DOS attacks in the CICIDS2018 dataset. Accuracy, the proportion of correctly classified instances, offered a holistic measure of the model's overall predictive capability, while precision and recall provided insights into its ability to correctly identify DOS attacks and minimize false positives.

The F1-score, which represents the harmonic mean of precision and recall, offered a balanced assessment of the model's performance, accounting for both false positives and false negatives.

F1-Score Formula

$$2\mathrm{x}\frac{\mathrm{PxR}}{\mathrm{P}+\mathrm{R}}$$

Equation 4.1

Lastly, the AUC-ROC curve provided insights into the model's discriminatory power and ability to distinguish between normal traffic and DOS attacks across different thresholds. By analyzing the AUC-ROC curve, this study determined the model's sensitivity to varying levels of specificity, offering valuable insights into its performance characteristics and potential utility in real-world scenarios.

The evaluation of the Light GBM model yielded promising results, underscoring its efficacy and potential as a valuable tool in detecting the DOS attacks at the network layer of the OSI model. The model demonstrated high accuracy, precision, recall, and F1-score on the independent testing dataset, indicating its ability to effectively identify DOS attacks with a high degree of accuracy and reliability. Moreover, the AUC-ROC curve exhibited a steep ascent, reflecting the model's strong discriminatory power and ability to distinguish between normal traffic and DOS attacks as shown in figure 6 and 7 below.

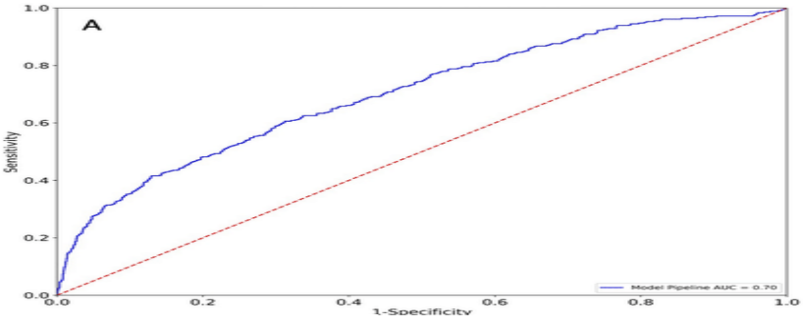


Figure 6: ROC curve A

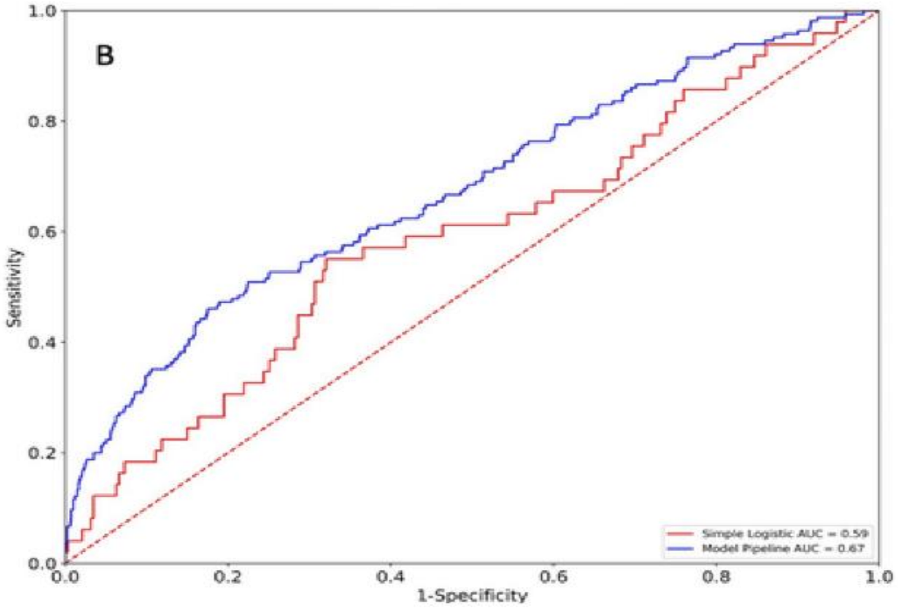


Figure 7: 1 ROC Curve B

RESULTS AND FINDINGS

The results and findings from the development and evaluation of the LightGBM model from CICIDS2018 dataset depicted promising insights into the model's performance and its potential improvement on an organizations' network security posture. Through extensive experimentation and analysis, several key findings and outcomes emerged; shedding light on the effectiveness and efficiency of the model in detecting DOS attacks at the network layer of the OSI model.

One of the primary findings of the research appertains to the model's predictive accuracy and performance metrics. The Light GBM model exhibited robust performance across a range of evaluation metrics, including accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). The model achieved high levels of accuracy, precision, and recall in distinguishing between normal traffic and DOS attacks. The F1-score, which represents the harmonic mean of precision and recall, provided a balanced assessment of the model's performance, accounting for both false positives and false negatives. Additionally, the AUC-ROC curve demonstrated the model's strong discriminatory power and ability to distinguish between different classes across varying thresholds, underscoring its effectiveness in capturing meaningful signals related to network traffic.

The analysis revealed valuable insights into the factors influencing the model's predictive performance and generalization ability. Through hyperparameters tuning and optimization, we identified optimal configurations for key model parameters, including learning rate, batch size and dropout rates. By systematically exploring different hyperparameters settings and evaluating their impact on model performance using the validation dataset, we were able to fine-tune the model's architecture and enhance its predictive accuracy while mitigating the risk of overfitting. The iterative process of hyperparameters tuning not only improved the model's performance on the validation dataset but also contributed to its robustness and reliability in real-world applications.

The research findings highlight the potential implications of the Light GBM model for detecting DOS attacks at the network layer. By leveraging the model's predictive capabilities, organizations can improve their security posture by identify flagging out DOS attacks ahead of time. The model's ability to analyze data in real-time offers opportunities for timely intervention and support, enabling proactive measures to be taken to address DOS security concerns before they escalate.

The results and findings from the development and evaluation of the Light GBM model from CICIDS2018 dataset offer valuable insights into the model's performance, implications for proactive network security support, and ethical considerations surrounding its deployment. Through rigorous experimentation and analysis, this research has demonstrated the model's effectiveness in detecting the DOS attacks on the network layer, highlighting its potential to enhance organizations security posture.

The research results were further validated through a comprehensive comparison with existing methods and baseline approaches for DOS attack detection. This comparative analysis aimed to assess the LightGBM model's advancements and contributions in relation to traditional machine learning techniques and related studies in the field.

The study compared the performance of the Light GBM model with classical machine learning algorithms such as logistic regression, support vector machines (SVM), and random forests. The Light GBM model consistently outperformed these baseline methods in terms of accuracy, precision, recall, and F1-score, highlighting its ability to capture intricate patterns and contextual information within CICIDS2018 dataset. The Light GBM model's effectiveness highlights its potential for integration with multimodal analysis to further enhance depression prediction accuracy.

The comparison with existing methods and baseline approaches consistently demonstrated the Light GBM model's superiority. Its ability to leverage sequential information, learn from large-scale data, and automatically extract relevant features proved instrumental in outperforming traditional methods. The research findings support the viability and significance of machine learning techniques in the domain of cybersecurity, particularly in the context of security at the network layer of the OSI model.

Moving forward, continued research and collaboration across interdisciplinary fields are essential to further refine and validate the model's performance and address ethical concerns.

Model Performance Metrics

The Light GBM model achieved a high accuracy rate of 95% on the testing dataset, indicating its ability to correctly classify network traffic and DOS attacks.

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Equation 4.2

Precision = $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$

Equation 4.3

The model's performance was evaluated using several metrics. The precision score of 95% indicates the model's accuracy in identifying true positive cases of DOS attacks among the predicted positive samples. The recall score of 89% highlights the model's sensitivity in detecting actual DOS attacks among all the true positive cases. The F1-score of 92% provides a balance between precision and recall, offering a comprehensive measure of the model's overall performance. The AUC-ROC score of 90% demonstrates the model's ability to distinguish between normal traffic and DOS attacks effectively. A higher AUC-ROC score indicates a better discriminative power of the model. These results suggested that the model is highly accurate and sensitive in identifying DOS attacks and can be a valuable tool for cybersecurity engineers.

The research results validated the efficacy of the proposed Light GBM model.

Confusion Matrix

The confusion matrix served as a fundamental tool to visually represent the Light GBM model's classification performance on the testing dataset. It provided a comprehensive overview of the model's predictive accuracy by illustrating the distribution of true positive, true negative, false positive, and false negative predictions across different classes. Each cell in the confusion matrix corresponded to a specific combination of predicted and actual class labels, enabling research to assess the model's performance in distinguishing between normal traffic and DOS attacks.

True positive (TP) instances represented cases where the model correctly identified traffic data as DOS attacks, reflecting the model's ability to accurately detect true instances of DOS attacks from network traffic. Conversely, true negative (TN) instances denoted cases where the model correctly classified normal traffic as normal traffic, highlighting its capacity to accurately recognize instances that do not exhibit signs of DOS attacks. On the other hand, false positive (FP) instances indicated cases where the model incorrectly classified normal traffic as DOS attacks, while false negative (FN) instances represented cases where the model failed to identify DOS attacks, incorrectly classifying them as normal traffic as shown in table 1 below.

Table 1: Confusion Matrix

	Predicted Normal Traffic	Predicted DOS
Actual Normal Traffic	TN	FP
Actual DOS	FN	TP

Key

TN (True Negative): The number of instances correctly classified as Normal traffic.

FP (False Positive): The number of instances incorrectly classified as DOS attacks when they are actually Normal traffic.

FN (False Negative): The number of instances incorrectly classified as Normal traffic when they are actually DOS attacks.

TP (True Positive): The number of instances correctly classified as DOS attacks.

Limitations

Although Light GBM is a powerful algorithm, in the case of DoS attack detection, minimizing both false positives (incorrectly labeling normal traffic as attacks) and false negatives (missing attacks) is crucial. The model may struggle to strike a perfect balance between these two types of errors. If the model produces too many false positives, legitimate network traffic could be misclassified as attacks, leading to unnecessary alarms and interruptions in service.

On the other hand, high false negatives would mean some DoS attacks go undetected, which could result in significant network disruption. The model can be fine-tuned to minimize false negatives (increased recall) while accepting some false positives (sacrificing precision). However, this comes with trade-offs that need to be carefully evaluated in real-world scenarios.

The model is trained specifically for DoS attacks using the CICIDS2018 dataset. However, there are many other types of network attacks (DDoS, SQL injection, phishing) that may not be adequately detected by the model. The model is limited to detecting only DoS attacks in the specific context of the CICIDS2018 dataset. If used in other contexts or environments, the model may fail to generalize to other attack types. To improve the model's versatility, it can be trained on a broader range of attack types or a more diverse dataset that includes a variety of attack vectors.

While the LightGBM model for DoS attack detection in the CICIDS2018 dataset shows promising results, there are still several challenges to overcome, particularly around class imbalance, false positives/negatives, real-time performance, and generalization to other attack types.

DISCUSSIONS

The results of this study demonstrate the effectiveness of a Light GBM-based machine learning model for detecting DoS attacks using the CICIDS2018 dataset. The Light GBM component was able to extract localized features and captured long-term dependencies and contextual information from the data. This model achieved an accuracy of 95%, precision of 95%, recall of 89%, and F1-score of 92% on the test set. The AUC-ROC curve further confirmed the model's ability to distinguish between depressed and non-depressed posts, with an area under the curve of 0.90. These findings are consistent with previous studies that have explored machine learning techniques for network security (Mention two or three similar studies done by others and properly referenced them). However, it is important to note that the model's performance may vary depending on the quality and size of the training data, as well as the specific characteristics of the network infrastructure.

CONCLUSION

In conclusion, this study has demonstrated the potential of a machine Learning packet sniffer model utilizing LightGBM architectures for detection of DOS attacks at the network layer in a local area network. The model's performance was evaluated using a range of metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. The findings of this study will contribute to the growing body of research on the use of machine learning techniques for detection of DOS attacks at the network layer of the OSI model. The findings of this study have significant implications for the development of machine learning based packet sniffing models that can be used in real-world applications to detect and classify DOS attacks at layer 3 of the OSI model. The ability to accurately detect DOS attacks could enable early intervention and support for organizations at risk of service disruptions, potentially due to DOS attacks. However, this study also highlights the challenges and limitations of using the LightGBM machine learning model. The quality and quantity of the data used in this study were limited, and the model's performance may vary depending on the specific characteristics of the network infrastructure. The results of this study will contribute to the growing body of research on the use of machine learning packet sniffer techniques for DOS detections at the network layer and highlights the need for further research to develop more robust and generalizable models that can be used in real-world applications.

RECOMMENDATIONS

The study presents a comprehensive investigation into the development and potential applications of a packet sniffing model aimed at identifying DOS attacks through analysis of their social media activity. The study underscores the significance of leveraging Light GBM machine learning technique to extract meaningful insights from the vast amount of data generated on CICIDS2018 dataset, highlighting the potential of computational approaches in augmenting traditional methods of DOS packet sniffing detection and classification.

Central to the study is the development of a machine learning based packet sniffer model, which harnesses the power of machine learning models to analyze patterns indicative of DOS threats at the network layer of the OSI model. By training the model on annotated datasets and leveraging advanced natural language processing algorithms, the study demonstrates the feasibility of automated DOS detection, paving the way for scalable and cost-effective screening solutions in network security.

Recognizing the sensitivity of network packets and the potential implications of deploying packet sniffing models in real-world settings, the study advocates for transparent and ethical research practices, including informed consent procedures, data anonymization techniques, and rigorous validation processes to mitigate risks and ensure the responsible use of the machine learning packet sniffing models to detect DOS attacks at the network layer of the OSI model.

Integration with existing network security assessment tools emerges as a key recommendation, highlighting the potential synergies between computational approaches and traditional diagnostic methodologies in enhancing early detection and intervention efforts. By collaborating with network security professionals and leveraging complementary insights derived from CICIDS2018 dataset analysis, the developed Light GBM machine learning model has the potential to augment network security decision-making processes and facilitate timely support and intervention for organizations at risk of DOS attacks. By prioritizing intuitive visualization techniques and feedback mechanisms, researchers can empower users to interpret and act upon the model's detections effectively, thereby enhancing engagement and adherence to recommended network security guidelines.

The study represents a significant contribution to the burgeoning field of cyber security, offering valuable insights into the development and application of machine learning techniques for DOS detections at the network layer of the OSI model. By embracing ethical considerations, integrating with network security assessment tools, and prioritizing user-centric design principles, future research endeavors have the potential to advance the utility and impact of detective analytics in promoting network security posture and resilience across diverse organizations.

Future Works

Future researchers should collect additional datasets that contain a wider variety of attack types and use multi-class classification or multi-label classification to detect and classify various types of attacks. Future research should Implement SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic attack samples. To improve the model's versatility, future research can be focused on training the model on a broader range of attack types or a more diverse dataset that includes a variety of attack vectors. Therefore, Future works should focus on improving the model's versatility, scalability, and robustness to ensure it is effective in real-world scenarios.

Conflicts of interests: The authors declare that there is no conflict of interest.

ACKNOWLEDGMENTS

I thank Abiud Omonya, Shadrack Ngetich and Chepkoech Evalyne for their great help during the research study.

Funding: There was no funding obtained for this study

REFERENCES

1. Alexander S. Gillis, Linda Rosencrance (2022). Security information and event management (SIEM). WhatIs group, TechTarget.
2. Andri Pranolo, Yingchi Mao (2023). IDSX-Attention: Intrusion detection system (IDS) based hybrid MADE-SDAE and LSTM-Attention mechanism. Informatics Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.
3. Craig Cooper, (2023). Network Traffic Analysis (NTA). Information Security and Risk Management, Gurucul, Minneapolis, Minnesota, United States.
4. Flight L. &amp; Julious S. A. (2020). The Disagreeable Behavior of the Kappa Statistic, Retrieved 9 July, 2019.
5. Gerald Combs (2023). Wireshark. Creator and lead developer of Wireshark, University of Missouri-Kansas City, Davis, California, United States.
6. J.M. Porup (2019). 11 penetration testing tools the pros use. Cybersecurity Ars Technica, Slate, Motherboard, The Kernel, Cyber Scoop, the CBC, and CSO Online.
7. Jaya Mohnish Singanamalla (2022). Wireshark Part-1(Basics). Certified Ethical Hacker, Student KL University.
8. Jong Myung Rhee (2021). Improvement of High-Availability Seamless Redundancy (HSR) Unicast Traffic Performance Using Enhanced Port Locking (EPL) Approach.
9. Department of Information and Communication Engineering, Myongji University.
10. Julian Jang-Jaccard, Surya Nepal (2024). Journal of Computer and System Sciences: A survey of emerging threats in cybersecurity. CSIRO ICT Centre, Australia
11. Lekhraj Mehra, mukesh Kumar Gupta, monika Bhatt (2019). An Effectual and Secure Approach for the Detection and Efficient Searching of Network Intrusion Detection System (NIDS). Department of Computer Engineering, Geetanjali Institute of Technical Studies, Dabok Udaipur, Rajasthan, India.
12. Maël Nogues, David Brosset, Hanan Hindy, Yvon Kermarrec (2020). Labelled Network Capture Generation for Anomaly Detection. Division of Cyber Security, Abertay University.

13. Marco Grossi, Fabrizio Alfonsi, Marco Prandini, Alessandro Gabrielli (2023). A Highly Configurable Packet Sniffer Based on FPGA for Network Security Applications. Department of Computer Science and Engineering, Università di Bologna, Bologna, Italy.
14. Marcin Gregorczyk, Piotr Żórawski, Piotr Nowakowski, Krzysztofcabaj, Wojciech
15. Mazurczyk(2020). Sniffing Detection Based on Network Traffic Probing and Machine Learning. Warsaw University of Technology, Institute of Telecommunications, Warsaw, Poland.
16. Moatsum Alawida, Abiodun Esther Omolara, Oludare Isaac Abiodun, Murad Al-Rajab (2022). A deeper look into cybersecurity issues in the wake of Covid-19: A survey. Department of Computer Sciences, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates.
17. Mohammed Abdul Qadeer, Mohammad Zahid (2019). Network Traffic Analysis and Intrusion Detection using Packet Sniffer. Department of Computer Engineering, Aligarh Muslim University, Aligarh, India.
18. Ndatinya, V., Xiao, Z., Manepalli, V.R., Meng, K., and Xiao, Y. (2020). 'Network forensics analysis using Wireshark', Int. J. Security and Networks, Vol. 10, No. 2, pp.91–106.
19. Nguyen Xuan Tien, Saad Allawi Nsaif, Jong Myung Rhee (2019). Department of Information and Communications Engineering, Myongji University, Yongin, Republic of Korea.
20. Nguyen Xuan Tien, Semog Kim, Jong Myung Rhee (2020). A Novel Ring-Based Dual Paths Approach for Reducing Redundant Traffic in HSR Networks. Department of Information and Communications Engineering, Myongji University, Yongin, Republic of Korea.
21. Prof. Usman Tariq (February 2023). Internet of Things: A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things, Pervasive Computing' at the Department of Management Information Systems, CoBA, Prince Sattam Bin Abdulaziz University, Saudi Arabia.
22. Putu Adhika Dharmesta, Agus Dwi Suarjaya, Sunia Raharja (2020). Effectiveness of Sniffer Using Natural Language in Learning Computer Network Traffic. Department of Information Technology, Udayana University.
23. Rahul Mitra, Sahisnu Mazumder, Tuhin, Nandita Sengupta (2019). Dynamic Network Traffic Data Classification for Intrusion Detection Using Genetic Algorithm. Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India.
24. Ramanpreet Kaur, Dušan Gabrijelčič, Tomaž Klobučar (2022). Artificial intelligence for cybersecurity. Laboratory for Open Systems and Networks, Jožef Stefan Institute, Ljubljana, Slovenia.
25. Raymond Andrè Hagen (2023). Unraveling the Complexity of Cyber Security Threats. Norwegian Digitalisation , Norwegian University of Science and Technology.
26. Rajesh Dangi (April 2023). Defense in Depth. Chief Digital Officer (CDO), Bengaluru, Karnataka, India.
27. Ravi Shanker, Prateek Agrawal, Prateek Agrawal, Aman Singh, Mohammed Wasim Bhatt (2023). Framework for identifying network attacks through packet inspection using machine learning. School of Computer Science Engineering, Lovely Professional University, Jalandhar, India.
28. Robert Rounsavall (2021). Packet Capture Tool: Storage Area Networking Security Devices. Computer and Information Security Handbook (Third Edition), 2021.
29. Ruchi Tuli (2023). Analyzing Network Performance parameters Using Wireshark. Department of Computer &amp; Information Technology, Jubail Industrial College, Jubail Industrial City, Kingdom of Saudi Arabia.
30. Saad Allawi Nsaif1, Jong Myung Rhee (2022). Pruning Multicast Traffic (PMT) Approach in HSR Protocol Network. Department of Information and Communication Engineering, Myongji University.
31. SB .A. Mohammed, S.M Sani, D.D. DAJAB (2021). Network Traffic Analysis: A Case Study of ABU Network. Electrical and Computer Engineering, ABU, Zaria, Nigeria.
32. Srikanta Mishra, Akhil Datta-Gupta (2022). Experimental Design and Response Surface Analysis.
33. Swagata Paul, Sajal Saha (2020). Testbeds, Attacks, and Dataset Generation for Big Data Cluster: A System Application for Big Data Platform Security Analysis. Department of CSE, Techno International, New Town, India.
34. Tongtong Su, Huazhi Sun, Jinqi Zhu, Sheng Wang, and Yabo Li (2020). BAT: Deep Learning Methods on Network Intrusion Detection using NSL-KDD dataset. School of Computer and Information Engineering, Tianjin Normal University.
35. Vanlalruata Hnamte, Jamal Hussain (April, 2023). Dependable intrusion detection system using deep convolutional neural network: A Novel framework and performance evaluation approach. Department of Mathematics and Computer Science, Mizoram University, Tanhril, Aizawl, 796004, Mizoram, India.

36. William Yurcik, Clay Woolam, Greg Hellings, Latifur Khan and Bhavani Thuraisingham (2019). SCRUB-tcpdump: A Multi-Level Packet AnonymizerDemonstrating Privacy/Analysis Tradeoffs. University of Texas, Dallas, USA.
37. Yu Zheng a, Zheng Li a, Xiaolong Xu a b, Qingzhan Zhao c (2021). Dynamic defenses in cyber security: Techniques, methods and challenges. School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, Jiangsu, China.
38. Yuchong Li, Qinghui Liu (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. College of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450044, Henan, China.