National College of Ireland

Optimizing Deep Packet Inspection for Securing Remote Work Communication Using Machine Learning: Addressing Performance & Privacy Concerns.

MSc Research Project
MSc Cybersecurity

Vaibhav Tupe
Student ID: X23162929

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Vaibhav Ramesh Tupe		
Student ID:	X23162929		
Programme:	MSc Cybersecurity	**Year:**	2024-2025
Module:	MSc Research Project		
Supervisor:	Niall Heffernan		
Submission Due Date:	12/12/2024		
Project Title:	Optimizing Deep Packet Inspection for Securing Remote Work Communications using Machine Learning: Addressing Performance & Privacy Concerns.		
Word Count:	8301 Words. **Page Count**: 27 Pages.		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Vaibhav Ramesh Tupe

Date: 12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Deep Packet Inspection for Securing Remote Work Communication Using Machine Learning: Addressing Performance and Privacy Concerns

Vaibhav Tupe

Student ID: x23162929

Abstract

The exponential rise in remote work in recent years has transformed organizational work fundamentally but it also created an escalating need for the cybersecurity need against the complexity and volume of such cybersecurity threats. Traditional network security techniques like Deep Packet Inspection (DPI) and others face many challenges in actually analyzing the encrypted traffic because of their inherent need for the attack signature patterns thereby reducing their detection against complex and covert cyber threats. This research study focuses on such limitations & challenges by the integration of traditional security measures with advanced Machine Learning (ML) techniques such that real-time traffic classification and threat detection is made possible in remote work environments as well as enterprise environments. This research study uses the comprehensive open-source dataset CICIDS 2017/2018 and various famous ML models including Random Forest, Support Vector Machine (SVM), and XGBoost. Using this dataset and these ML models will accurately classify major network traffic as benign or malicious based on the encrypted data and only using protocol-specific metadata which is extracted using the nDPI library. In the networking world there are many strict privacy standards which must be met and as such this research uses the Encrypted Traffic Analysis (ETA) to ensure user data privacy. This research study utilizes Docker for simulating remote work network environments and Elastic Stack for real-time logging and visualization. Empirical results showed that this proposed DPI-ML framework has performed with exceptional classification accuracy and has also maintained low latency and better throughput while not compromising on the user data privacy. This study was also tested in comparative benchmarking against the existing DPI and ML-based solutions and the results highlight better performance as it not only advances the abilities of traditional DPI but also provides a scalable solution tailored to the dynamic security needs of any remote work environments.

Keywords: Remote Work, Deep Packet Inspection, Machine Learning, Network Security, Encrypted Traffic Analysis, Real-Time Classification, Privacy Preservation

1. Introduction

In the digital age, the transformations for the global events and the technologies have skyrocketed the research and have reshaped how various organizations of the world operate. This has been especially true with remote work becoming more and more famous. But this also presents many security and networking challenges which the new world is not yet prepared for. The need for the secure and the reliable communication for such remote work and in-general internet communication is all the more vital in this day and age, especially in the distributed work environments where sensitive organizational data has to be transmitted over longer ranges of the world. In this regard, cybersecurity has many challenges to overcome and creating a performance network which works well against many types of surface attacks responsible for remote work issues. Traditional network security approaches are insufficient for such long-range secure internet communication and need the use of various Machine Learning techniques to make the communication more robust and attack proof. In such a context, Deep Packet Inspection (DPI) has been an invaluable tool for the network security against various emerging threats like cyber data breaches, unauthorized access, and various network exploitation. DPI offers an in-depth analysis for the packet headers and the vast metadata such as the analysis of the malicious activity in real-time.

There are many benefits of using the DPI but similarly there are many limitations to it too. The particular limitation regarding our research is the increased usage of the encrypted communication channels and their privacy concerns. In today's world of internet and remote work, the traffic is encrypted which has limited the DPI's effectiveness as it is unable to inspect with optimal efficiency as it must decrypt before the data analysis. This in turn produces many privacy risks. Furthermore, the traditional DPI techniques/methods are very resource intensive, latency inducing, and they bottleneck the network systems at large, even in under the high traffic demand. In order to address these limitations and solve the various cons of using the DPI, the research community has followed the Machine Learning paradigm which promises the enhancement of DPI. Combined with ML models, the various patterns inside the network attacks and the malicious activities on the packet metadata, is a promising start. This approach will help maintain the security concern alleviation while also offering the direct balance between the robustness of the threat detection and also the optimal performance impact.

In the research community an indirect competitor to machine learning is the advent of specific privacy preserving centralized deep learning methods which has seen much popularity and rise in network security as a privacy preserving alternative. It gives its contribution to its decentralized approach instead of the machine learning's centralized approach which could be beneficial depending on the network architecture and the types of the network attacks which needed to be countered. In this research, such a machine learning model's distributed approach also finds much credence, as the goal of this research is closely aligned to secure remote communication, minimization of the data aggregation, and thus reducing the privacy attack risks. In the context of the DPI application, it has a resounding success in the continuous refinement of the various threat detection models. This could be useful in the countering of the various network threats out there and how to minimize those threats in the live distributed network environment.

A hybrid approach between the usage of the ML has been the raging debate of the research community at large when solving the network's vast distributed nature but also the need for the centralized attack detection/solution of the various types of the network attacks. For example the usage of the DPI and the study by the Lotfollahi et al. (2020) introduced a deep learning based framework which is referred to Deep Packet in their study, as it achieved a better performance, higher accuracy compared to the traditional ML, and the identification over edge of the traffic patterns compared to the CNNs. Similarly the famous open source library nDPI has become the foundational tool to be used in the high speed DPI which has demonstrated the capability of the protocol classification at the packet level without payload inspection and similar performance enhancement metrics. nDPI library also majorly combines the various ML classifiers like Random Forest and SVM, in order to combat the real time traffic analysis of the high performance network environments.

There is also a need to elaborate that the existing studies by the research community have still not fully addressed the unique combination nature of the various privacy preservation, low latency and high security, which may be needed for remote work communication. Moreover the integration of the real time DPI with various ML models while also tackling the optimization of the privacy and performance both, have remained a challenge still in the research community. This is especially true in the encrypted traffic environment of today's network industry. This study builds itself on the groundwork laid by all the previous research as it proposes the new and innovative architecture which combines the nDPI based packet inspection with ML models trained on the open-source datasets such as CICIDS 2017/1018 for real time classification on the network traffic in the remote work environments. This research proposes a system which uses the ML to train a model locally to enhance the user's privacy and their security while also combating the encrypted traffic.

1.1. Research Question

How can Deep Packet Inspection (DPI) be optimized using Machine Learning (ML) techniques to secure remote work communications, while not compromising the performance and privacy concerns?

The main objectives of this research are as follows:

- Optimize the deep packet inspection for real-time traffic classification
- Incorporate privacy-preserving methods
- Evaluate the system performance across multiple metrics

In this research study we first implemented a simulated remote work environment using the docker tool while also simulating the network traffic which would be monitored, logged, and classified in real time with various such tools. This system architecture will use the nDPI while also use the combination of the ML models like the Random Forest, SVM and XGBoost for real-time classification of both the benign and the malicious traffic.

2. Related Work

In today’s world of remote connection supremacy, secure connections have become very important in such a small span of internet lifetime. Especially given the fact that the increase in the remote network traffic and the rise of its cybersecurity risks are deeply intertwined. Deep Packet Inspection (DPI), traditionally was used for the networking monitoring and the data analysis, but it has gained more popularity in recent times and has also garnered enough attention for being a robust mechanism for all kinds of network threat detection. This network threat detection is done by using the analysis techniques on the network traffic at a very low level i.e. granular level (Hypolite et al., 2020). But as the encryption techniques continue to evolve and the privacy issues become even more serious than what they are today, the need for enhancing the DPI through Machine Learning will be very effective as it has shown extreme promise in other areas of the Machine Learning paradigm (Song et al., 2020).

In this section of the research, the following literature review will dive into the various advances in the DPI and ML integration and their applications for privacy preservation, and the various performance enhancement techniques used by the research communities throughout the world. In the context of the Deep Packet Inspection (DPI), recent research has shown how effective they are when extracting information and inspecting data in both the packet headers and the metadata present. This has some extreme benefits for the detection of the malicious activity without the decryption required of the sensitive data (Deri and Fusco, 2021). The famous nDPI library represents a very important and turning point for the development in the open-source DPI for high-speed packet inspection and protocol classification as discussed by Deri and Fusco (2021). This has shown great promise in the various network monitoring and the networking projects providing the framework which balances speed and accuracy for the real-time analysis of the network in the subject remote work environments.

There are more advancements in the DPI detection capabilities and they are gained by using/integrating the ML models alongside the DPI framework. Kim et al., 2021 proposed that the Deep Packet framework can be used alongside the various deep learning models such as CNNs or stacked auto encoders to easily classify even the encrypted traffic. This framework is especially useful where the total packet payload is not known or inaccessible. This framework relies on pattern recognition to identify the various potentially malicious activities present inside. With these kinds of approaches, ML's full potential to improve the DPI by automation of the feature extraction even inside the encrypted data is impressive and highly useful.

In the traditional DPI the biggest concern or cons is that they compromise the user’s privacy. This is especially true during the inspection of the remote sensitive work and their corresponding communications. Machine learning comes to the rescue to solve such a compromise by proposing a solution which mitigates the privacy risks. This privacy mitigation is achieved by various ML models trained on the distributed decentralized nodes without the collection of the raw data in a singular place (Nkongolo et al., 2022). In the context of his research, Nkongolo et al., 2022 also proposed a personalized federated multi-task learning scheme to address the encrypted data’s heterogeneity in various organizations. Thus the technique proposed by Nkongolo et al., 2022 allows the various numerous enterprises to train powerful ML models without invading the user’s privacy data.

In a similar study Nyasore et al., 2020, showed that machine learning can be expanded by proposing another secure federated distillation framework. This framework creates a model distillation which greatly reduces the overhead caused by the communication and increases the potential accuracy in contrast to the traditional ML approaches. In the decentralized systems, this model performs best as its technique is especially useful in the real-time DPI remote work with the emphasis on the individual's privacy.

Sainz et al., 2020 built on such advancements and tackled the problem of the user's privacy and the thorough inspection of the various deep network packets by creating a hypernetwork-based machine learning framework. This network classified the encrypted traffic by further enhancing the privacy and personalization of the DPI systems by adapting the model parameters to local traffic patterns.

The analysis of the internet traffic which is encrypted remains a challenge still and it is especially difficult in the context of the DPI where deep packet inspection is necessary for the remote network communication safety. Traditional DPI methods fall short and are often limited in their inspection of the metadata of the network traffic. This is due to the encrypted nature of the data that is being passed through, but recent advancements have explored a high volume of the encrypted sophisticated traffic. Lu et al., 2021 showed a combination of machine learning with homomorphic encryption. This resulted in such traffic analysis, where the data, although encrypted, is analyzed without exposing i.e. decryption. This approach is very important as it helps solve the consistent issue in the deep packet analysis of the encrypted metadata. Lu et al., 2021 research "SlimBox: Lightweight Packet Inspection Over Encrypted Traffic" extends this concept even further. It does so by introducing the concept of the secure keyword-based DPI which allows the network administrators to quickly identify threats without ever exposing the sensitive data and violating the user's privacy. This can be especially good for remote network traffic communication where the user's privacy is a big concern. Moreover, Ning et al. (2020) in their research proposed a protocol which could enhance the theoretical accuracy of the DPI over TLS (Transport Layer Security) by integrating the implementation of rule hiding and dynamic rule addition. This not only preserved the user's privacy but also allowed the DPI to thoroughly inspect the data inside the packets, thus making it an even better alternative to the traditional DPI.

In the context of the real-time DPI remote work communication, a very delicate balance is needed between both security and the network performance, as either of them decreases when the other increases. Traditional DPI methods fall short because of their latency induction which arises because of the extensive analysis of the packet headers and the metadata. But recent studies have shown that such a delicate balance where neither security nor network performance gets affected is possible and it is achievable with the Machine Learning techniques.

The nDPI library proposes itself to be the most important factor in both optimizing DPI network inspection speed and also network security. This is done by using both the packet headers and their relevant payloads for fast protocol classification where the real-time analysis enables it without any further delays (Deri and Fusco, 2021). One of the examples of such is the integration of the ML models like Random Forest and SVM, where nDPI can perform efficient remote network traffic communication classification. This is also true for the environments where the throughput is high.

Further studies, such as Song et al. (2020) showcase the role of DPI in cybersecurity detection. This is done by using security scores derived from the various ML models. These security scores then in turn identify malicious patterns in the remote network traffic. These methods perform best and can even minimize latency while guaranteeing a comprehensive security coverage. Moreover, another study such as Alkhalidi and Yaseen (2021), shows that integrating social network analysis can greatly enhance the performance because they can identify patterns more quickly which makes the redundant processing time reduced.

While DPI, the ML integration, and privacy-preserving techniques, all show an impressive amount of advancements there are still many more challenges ahead. The most common challenge which still remains is the encrypted traffic. As this encrypted traffic presents a lot of limitations in DPI capabilities, thus needing complex cryptographic techniques which can really slow down real-time performance. Future research should explore lightweight encryption analysis models and adaptive ML techniques.

3. Methodology

The methodology section in this research study shows the systematic approach used to achieve the research objectives of optimizing Deep Packet Inspection (DPI) for securing remote work communications using Machine Learning (ML) techniques. This section is about the research design, data acquisition and preprocessing, feature extraction, model development and training and system integration. Each sub-section provides in-depth details of the processes which were involved and techniques which were used.

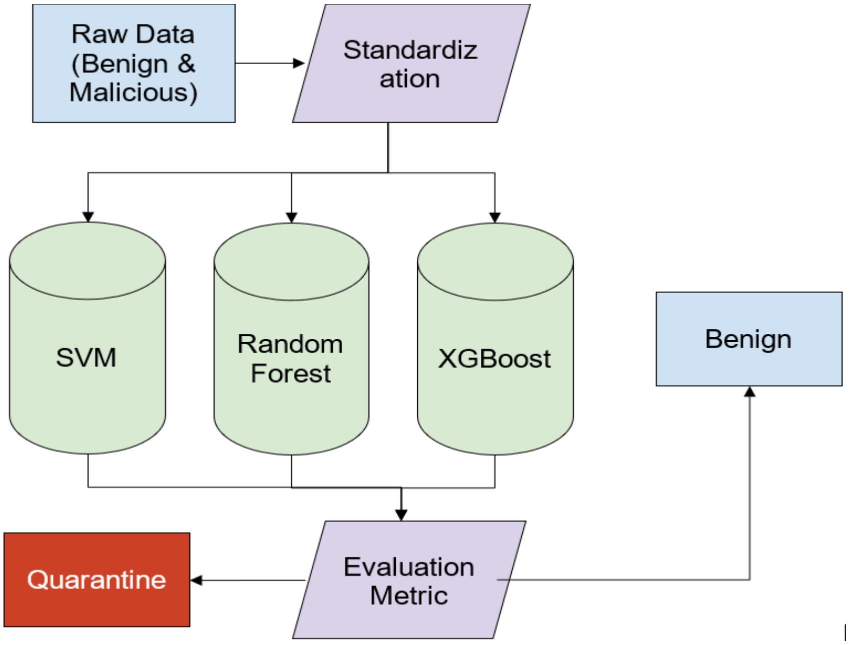


Figure 1: Methodology Overview of the Study for Benign and Malicious Actors

This figure outlines the research steps, showing how data moves through the system and it does so by highlighting the key actions. Each step has a clear purpose, such as improving the data quality or enabling accurate classification which will ensure a reliable system performance. The overall methodology depends upon a number of factors including the following classification tools which can accurately differentiate between the benign and the malicious actors:

Step	Action	Purpose	Tools/Technologies
Data Collection	Use the CICIDS 2017/2018 dataset for benign and malicious network traffic patterns.	Provide a comprehensive and labelled dataset for ML model training.	CICIDS Dataset
Data Cleaning	Remove duplicates, handle missing/infinite values and standardize labels.	Ensure data quality and consistency for reliable model training.	Pandas, NumPy
Data Balancing	Apply under-sampling (benign class) and SMOTE (malicious class).	Address class imbalance to prevent biased model training.	Imbalanced-Learn (RandomUnderSampler, SMOTE)
Feature Extraction	Extract metadata such as packet size, timing, and protocol details using nDPI.	Enable classification without accessing encrypted payloads, ensuring privacy.	nDPI Library
Feature Selection	Select features with high correlation and low interdependence; normalize	Reduce dimensionality and enhance model accuracy.	Scikit-Learn (Feature Importance, MinMaxScaler)

	feature values.		
Model Selection	Evaluate Random Forest, SVM and XGBoost models for classification.	Leverage diverse model strengths for accurate traffic classification.	Scikit-Learn, XGBoost
Model Training	Train models on preprocessed, balanced data; optimize hyperparameters with Grid Search.	Develop accurate and robust classification models.	GridSearchCV, Cross-Validation
System Integration	Embed trained models into the DPI system for real-time classification.	Enable real-time traffic monitoring and threat detection.	Python Scripts, Joblib/Pickle
Privacy Preservation	Analyze encrypted traffic metadata without decryption; use decentralized model training.	Maintain user privacy and comply with data protection regulations.	Encrypted Traffic Analysis (ETA), Federated Learning
Environment Simulation	Use Docker to emulate realistic remote work network conditions; simulate benign and malicious traffic.	Validate system performance under diverse and dynamic conditions.	Docker, iperf, hping
Real-Time Logging	Monitor classification results and traffic patterns using Elastic Stack.	Provide actionable insights and visualization of system performance.	Elastic Stack (Logstash, Elasticsearch, Kibana)

The methodology section outlines the practical steps we have thus far taken to achieve the research objectives and also focused on the data preparation, feature selection and the integration of advanced tools and models and this also explains how we used a comprehensive dataset, balanced the data for fairness, and extracted meaningful features using cutting-edge tools like the nDPI library which will help later in this section where we highlight the training of machine learning models to classify network traffic and the implementation of privacy-preserving techniques. This foundation sets the stage for integrating these components into a scalable system which is also tested in realistic environments to ensure its efficiency and reliability.

3.1. Research Design

This research uses an experimental and quantitative design in which advanced DPI techniques are combined with ML models to enhance the subject network security in remote work environments while not deviating from the primary focus which is on developing a robust system able to classify network traffic in a real- time environment while also guaranteeing the optimal performance and preserving the user privacy. This research follows a structured and iterative approach i.e. data collection, preprocessing, feature extraction, model training, system integration and comprehensive evaluation because such a design helps the exploration between DPI, ML and privacy-preserving techniques.

3.2. Data Collection and Preprocessing

The cornerstone of this research is the dataset called the CICIDS 2017/2018 which is famous for its comprehensive and well-labelled network traffic data and it is common for such datasets to have lots of data for benign and malicious traffic patterns which makes them ideal for training and evaluating ML models in such traffic classification tasks.

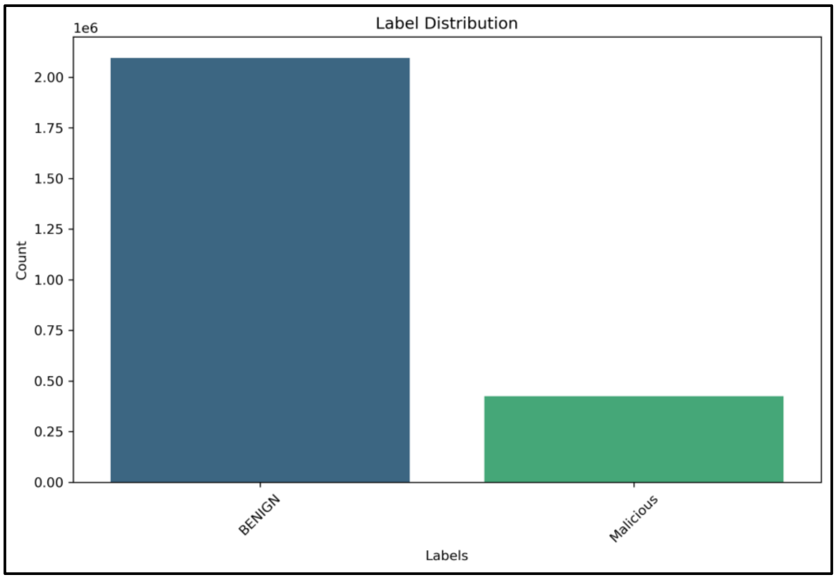


Figure 2: Illustration showing the Label Distribution (Benign and Malicious)

This chart displays the number of benign and malicious samples in the dataset. It helps visualize the class imbalance before processing. Balancing the data ensures the machine learning model doesn’t favor one class over the other, leading to fairer and more accurate results. The CICIDS datasets provide very fine details on network packets like headers and metadata making it a must have for any effective DPI and ML integration. The raw CICIDS datasets go through cleaning and preparation steps to ensure data quality for ML model training and the following steps are taken:

- Utilizing Pandas isnull() function to detect columns with missing values.
- Dropping rows with null values to prevent skewed model training.
- Using Pandas duplicated() method to identify duplicate rows.
- Removing 307,376 duplicate rows to ensure that each data point is unique.
- Scanning for positive and negative infinity values across the dataset.
- Converting relevant values to NaN using NumPy's replace() function.
- Dropping rows containing NaN values post-replacement.
- Identifying columns with a single unique value using Pandas' nunique() function.
- Eliminating 8 constant columns (bwd_psh_flags, bwd_urg_flags, etc.) to reduce dimensionality.
- Simplifying the label column to binary categories i.e. BENIGN and MALICIOUS.

3.3. Data Balancing

Addressing class imbalance is a very important step to prevent biased model training and as such the following techniques are employed to achieve a balanced dataset:

- Utilizing RandomUnderSampler from the imb learn library to reduce the number of benign traffic instances to half their original size.
- Reduce the benign traffic instances from 2,099,057 to approximately 1,051,484.
- Applying Synthetic Minority Over-sampling Technique (SMOTE) via the SMOTE class from the imblearn library.

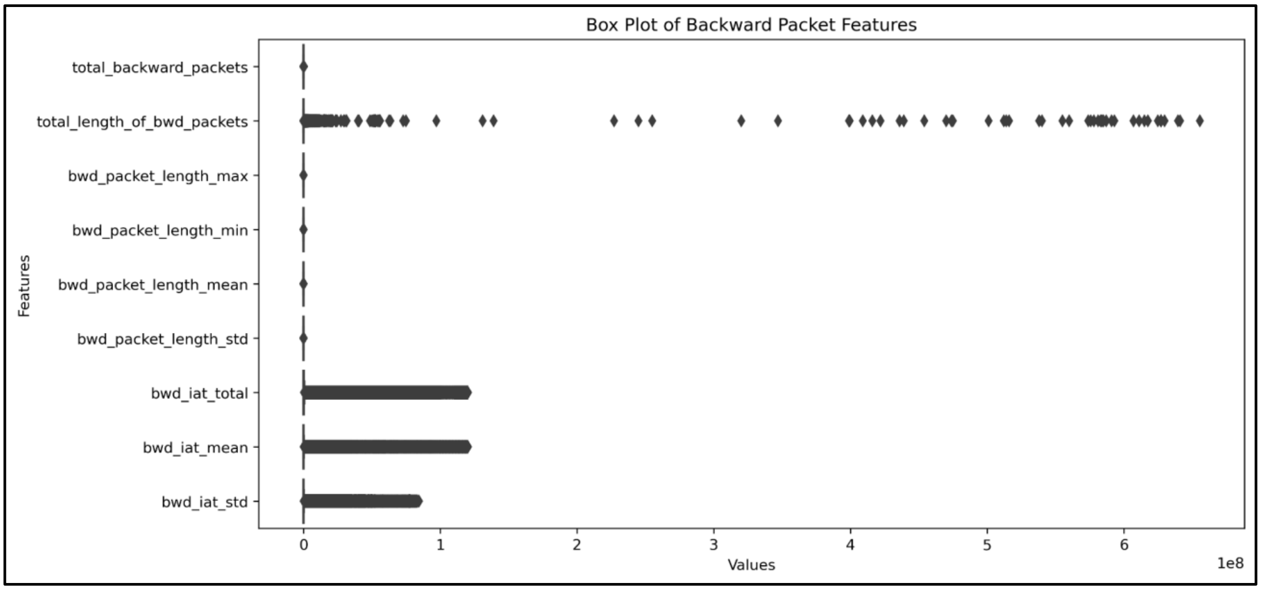


Figure 3: Graph showing the box plot of the backward packet features

This plot represents the distribution of certain packet features. It helps identify outliers and patterns within the data, ensuring features are properly understood and prepped for training machine learning models.
Clean and consistent data is critical for reliable predictions.

- The minority class is over-sampled which results in a perfectly balanced dataset with 851,482 instances each for benign and malicious traffic.
- The combined balancing approach yields a dataset of 1,702,964 rows with an equal distribution of benign and malicious traffic.
- Splitting the balanced dataset into training and testing subsets.
- Ensuring that the split maintains the class distribution by using the stratify parameter.
- Applying StandardScaler from scikit-learn to normalize feature values.
- Fitting the scaler on the training data and transforming both training and testing datasets.

3.4. Feature Extraction and Selection

The nDPI library is used for extracting protocol-specific metadata from network packets and thus by integrating nDPI with the ML models the system can classify network traffic based on protocol patterns without compromising any of the encrypted payloads and user privacy and so the extraction process involves:

- nDPI analyzes packet headers to identify the underlying protocol (e.g., HTTP, HTTPS, FTP) used in the communication.
- Extracting features such as packet size, inter-arrival times, flow duration and other header-specific

attributes.

- Selecting features which have high correlation with the target variable (label) and low inter-correlation.

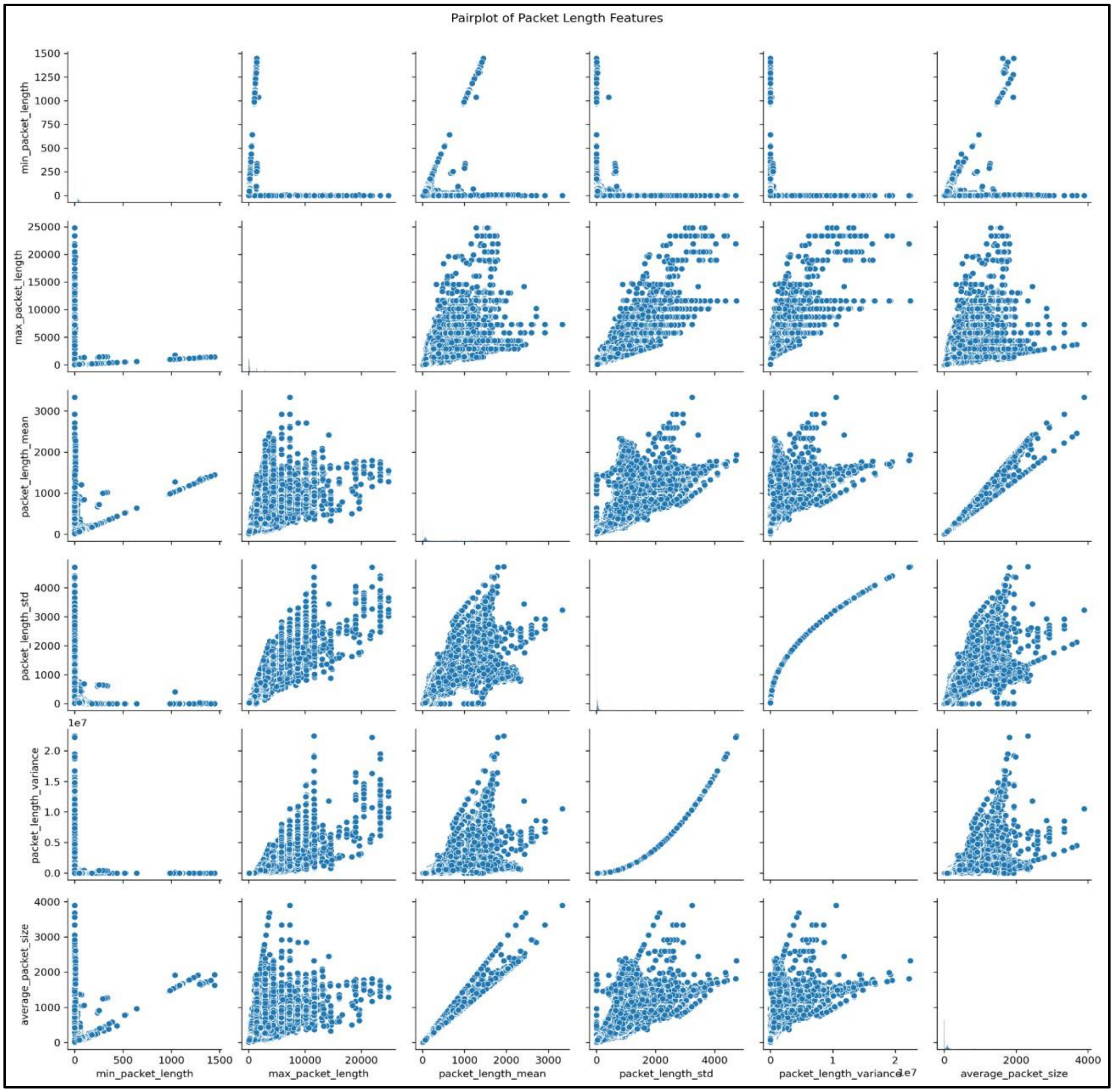


Figure 4: Collection of charts depicting the pairplots of packet length features

These charts show relationships between packet length features and help identify correlations. Strong correlations indicate which features are useful for classification, simplifying the data and improving model performance.

- Using Pearson correlation coefficients and feature importance scores.
- Features are scaled to a uniform range using MinMaxScaler and StandardScaler.
- Applying techniques like one-hot encoding or label encoding to convert categorical features into numerical representations for ML models.
- Principal Component Analysis (PCA) for dimensionality reduction.

The nDPI library was chosen for metadata extraction due to its great ability to analyze the said encrypted traffic headers without causing the payload to any decryption process. Unlike the typical payload-based DPI methods which only compromise with the user's privacy, this nDPI method focuses entirely on the protocol-specific metadata which makes sure that the compliance is achieved with any privacy regulations for the user’s security while maintaining the high classification performance. This also aligns with the research's objective to balance privacy and threat detection.

3.5. Integration of ML Models with DPI Framework

The integration of ML models with the DPI framework ensures a seamless data flow, real-time processing, and minimal latency such that the following components are proposed for the architecture:

Aspect	**Details**	**Key Features**
DPI Framework	Real-time packet inspection and metadata extraction	Protocol classification using nDPI
ML Models	Random Forest, SVM, XGBoost classify traffic	Detects benign vs. malicious traffic
Performance Goal	Low latency, high throughput	Real-time analysis focus
Optimization	Efficient pipelines, parallel processing	Speed and resource optimization
Logging & Monitoring	Elastic Stack (Logstash, Elasticsearch, Kibana)	Real-time classification insights
Environment Simulation	Docker for realistic testing	Controlled, scalable setup
Setup	Configure nDPI for live traffic inspection	Tailored to ML model needs
Integration	Embeds trained ML models into DPI framework	Immediate classification
Automation	Scripts for automatic feature feeding	Streamlined classification workflow
Data Handling	Efficient pipelines reduce processing time	Faster inspection-to-classification
Resource Management	Adequate provisioning in Docker	Handles high traffic without performance drop
Scalability	Supports traffic load variations	Scalable architecture
Parallel Processing	Multi-threading for real-time performance	Enhanced throughput
Visualization	Kibana dashboards for insights	Traffic patterns, system metrics

Three ML models i.e. Random Forest, SVM, and XGBoost are specifically selected to classify network traffic and among them the Random Forest model was chosen because of its robustness against such overfitting issues and also because of its ability to handle such large feature sets. SVM on the other hand was chosen for the very strong performance in dealing with the high-dimensional spaces, while XGBoost was used for its computational efficiency and superior accuracy in the said capture of the high and complex patterns in such a way that these models could complement each other and the nDPI framework thus making sure to be accurate and do some real-time classification.

4. Design Specifications

The Design Specifications section shows the architectural blueprint and technical components essential for developing an optimized Deep Packet Inspection (DPI) system combined with Machine Learning (ML) techniques such that the system architecture, data processing pipelines, integration mechanisms for ML models and privacy-preserving methodologies are discussed here. Additionally it also includes various relevant tables to better visualize and help understand.

4.1. System Architecture

The system architecture is crafted in such a way so as to facilitate the integration between DPI functionalities and ML-based traffic classification while ensuring better performance and better privacy and thus this architecture is modular and it comprises various components which analyze real-time traffic and monitor any threat detection. The primary steps of the system architecture are as follows:

Component	**Function**	**Key Benefit**
Packet Inspection (nDPI)	Real-time packet/metadata capture without payload access	Preserves privacy, supports ML analysis
Metadata Preprocessing	Normalize, encode, and refine features for ML input	Ensures consistent, high-quality data
ML Classification	Classify traffic (Random Forest, SVM, XGBoost)	Dynamic threat detection
Decentralized Training	Train models locally without centralizing raw data	Enhances privacy and regulatory compliance
Encrypted Traffic Analysis	Apply keyword-based DPI to encrypted metadata	Detects threats without decrypting payloads
Low-Latency Execution	Minimize delay from capture to classification	Maintains real-time performance
Logging & Visualization (Elastic)	Real-time logging, indexing and dashboarding	Provides actionable monitoring and insights
Docker Simulation	Emulate diverse, realistic network conditions	Robust testing and validation

4.1.1. Data Flow and Processing Pipeline

The data flow within the system architecture is made through a streamlined processing pipeline designed to handle high-volume traffic. The data flow and processing pipeline for this proposed DPI-ML system involves several critical stages. These stages include a packet interception technique using nDPI, which extracts the headers and metadata from the captured network packets for use in ML algorithms. The raw metadata output consists of information such as protocol type, size and timing. Next, normalization scales these numeric features to a consistent range of features and their encoding converts this categorical data into a numeric format making it usable by our machine learning models. Feature selection focuses on retaining only relevant attributes to improve model accuracy. To address the class imbalance, our system uses under-sampling for benign samples and SMOTE for synthetic minority over-sampling of malicious samples. Metadata analysis through nDPI focuses on the protocol-level details to further enhance classification quality, while derived features add pattern-based attributes to improve overall performance. The training phase involves fitting machine learning models on the processed and balanced data, followed by hyperparameter tuning to achieve optimal accuracy and stability.

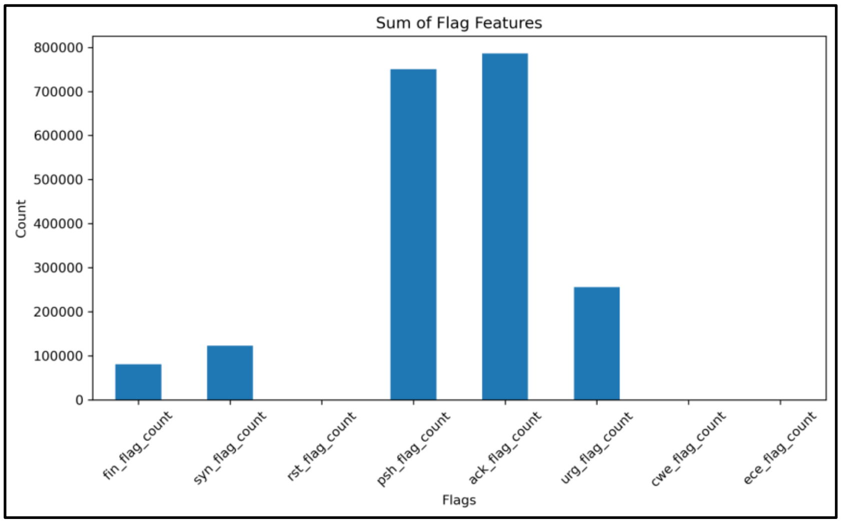


Figure 5: Bar chart showing the sum of flag features and their count

This chart highlights how often specific packet flags appear in the dataset. These flags are important indicators of traffic behavior, helping distinguish between benign and malicious activity.

4.2. Integration of ML Models with DPI

The integration of ML models with the DPI framework is very important for achieving real-time traffic classification with high accuracy and minimal latency and the following integration process helps:

Component	**Description**	**Role**
Packet Capture Module	Uses nDPI to capture and inspect packets in real-time, extracting headers and metadata only	Provides essential traffic data for analysis
Data Preprocessing Engine	Normalizes, encodes and selects features from raw metadata	Ensures consistent, high-quality input for ML

Machine Learning Models	Uses Random Forest, SVM and XGBoost to classify traffic	Identifies benign vs. malicious activity
Encrypted Traffic Analysis Module	Examines encrypted metadata without decryption to detect threats	Preserves privacy while identifying risks
Real-Time Processing Unit	Coordinates rapid data flow from capture to classification	Maintains low latency and high throughput
Logging & Visualization Interface	Uses Elastic Stack for real-time logging and dashboards	Enables monitoring and actionable insights
Simulation Environment	Employs Docker to simulate realistic network conditions	Facilitates thorough testing and refinement

This table shows that the DPI-ML system consists of several integrated components. These are designed for efficient, privacy-preserving, real-time traffic classification in remote work environments and this Packet Capture Module uses the nDPI to extract the headers and metadata from network packets without accessing encrypted payloads. The Data Preprocessing Engine normalizes, encodes and selects relevant features for the machine learning models like Random Forest, SVM and XGBoost to be used as input. This Encrypted Traffic Analysis Module examines the encrypted metadata without decryption. After that the Real-Time Processing Unit makes sure that data is rapidly flowing with minimal latency and high throughput. The Logging & Visualization Interface uses the Elastic Stack to log and visualize system performance. These steps are performed iteratively for the Machine Learning model's integration.

4.3. Privacy-Preserving Mechanisms

Ensuring user privacy is very important for the DPI-ML framework in remote work environments where sensitive data is frequently transferred and so the system uses the advanced privacy-preserving techniques to reduce the privacy risks.

Step	**Description**	**Tools/Technologies**
Packet Capture	Captures and inspects network packets, extracting headers and metadata without payload access.	nDPI, Wireshark
Data Cleaning	Removes missing values, duplicates and handles infinite values to ensure data quality.	Pandas, NumPy
Feature Selection	Identifies and retains relevant features correlated with the target variable.	Scikit-Learn, Feature Importance
Data Balancing	Balances the dataset using under-sampling and oversampling techniques to address class imbalance.	Imbalanced-Learn (RandomUnderSampler, SMOTE)
Feature Scaling	Normalizes feature values to ensure uniformity across the dataset.	Scikit-Learn (StandardScaler)

Model Training	Trains Random Forest, SVM and XGBoost models on the preprocessed and balanced dataset.	Scikit-Learn, XGBoost
Hyperparameter Tuning	Optimizes model parameters using grid search and cross-validation to enhance performance.	GridSearchCV, Cross-Validation
Model Integration	Embeds trained ML models into the DPI framework for real-time classification.	Python Scripts, Joblib/Pickle
Real-Time Inference	Classifies live network traffic based on extracted metadata using integrated ML models.	nDPI, ML Models
Logging and Visualization	Logs classification results and visualizes traffic patterns and system performance metrics.	Elastic Stack (Elasticsearch, Logstash, Kibana)

5. Implementation

This implementation section shows the many steps taken to create the proposed Deep Packet Inspection (DPI) system combined with Machine Learning (ML) techniques for the security of remote work communications as it consists of the environmental setup, dataset preparation, ML model development and training, system integration and privacy-preserving techniques. Each phase provides a clear roadmap for future research and deployment and is as follows.

5.1. Environmental Setup

The successful deployment of the DPI-ML system shows that it is a robust and scalable environment which can simulate real-world remote work network conditions and also provides integration of various components and it does so by the environmental setup which consists of the installation and configuration of essential software tools, libraries and platforms etc.

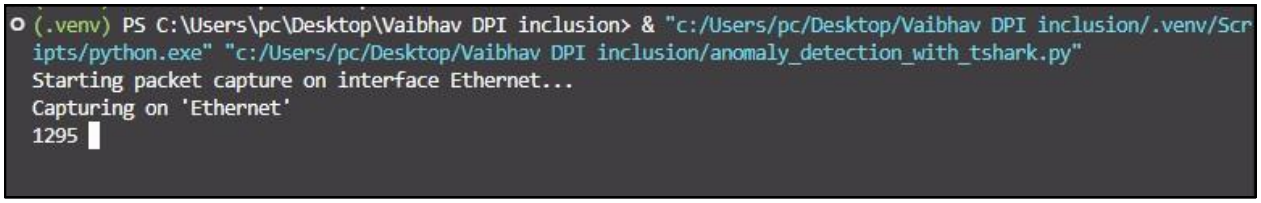
5.1.1. Tool Installation and Configuration

To simulate a realistic remote work environment and to also test the proposed DPI-ML framework the following steps are done:

Tool/Technology	**Purpose**
Docker	Isolated containers for testing
nDPI	Real-time packet inspection
Wireshark	Traffic capture and analysis
Elastic Stack	Logging, monitoring, visualization
Python (venv)	Manages ML libraries and data tasks

5.1.2. Network Topology Simulation

A Docker-based simulation environment was created so that it can be exactly like the typical remote work network as it contains various components such as virtual private networks (VPNs), software-defined wide area networks (SD-WANs), and cloud-based systems.



```
○ (.venv) PS C:\Users\pc\Desktop\Vaibhav DPI inclusion> & "c:/Users/pc/Desktop/Vaibhav DPI inclusion/.venv/Scr
  ipts/python.exe" "c:/Users/pc/Desktop/Vaibhav DPI inclusion/anomaly_detection_with_tshark.py"
  Starting packet capture on interface Ethernet...
  Capturing on 'Ethernet'
  1295 █
```

Figure 6: Figure representing the tshark working to gather packets

This figure shows how packet data is captured in real time using tools like tshark. It illustrates the initial step in data collection for the system, where raw network traffic is converted into usable data for analysis. The simulation consists of the following steps:

- Multiple Docker containers were created to represent different network nodes, including clients, servers and gateways and as such each container was assigned unique network configurations.

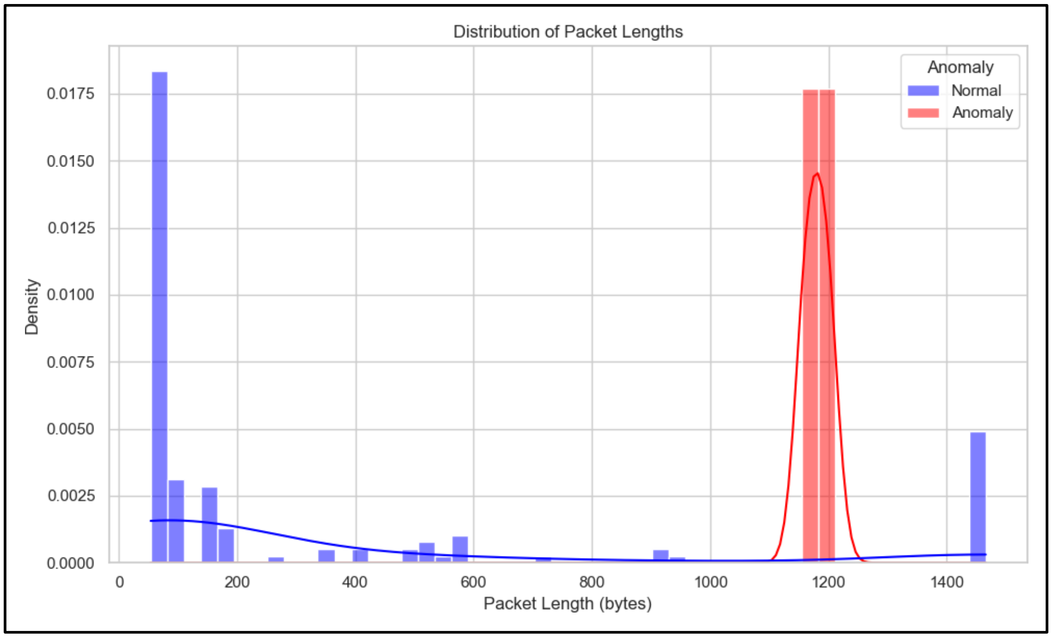


Figure 7: Network chart showing the captured packet's distribution of packet length

This network chart displays the distribution of packet sizes, providing insights into traffic characteristics. It is a foundational step in understanding the dataset and preparing it for feature extraction and model training.

- Tools like iperf and hping were used within Docker containers to generate benign and malicious network traffic.
- The simulated network traffic is directed via the DPI module powered by nDPI for real-time packet inspection and metadata extraction.

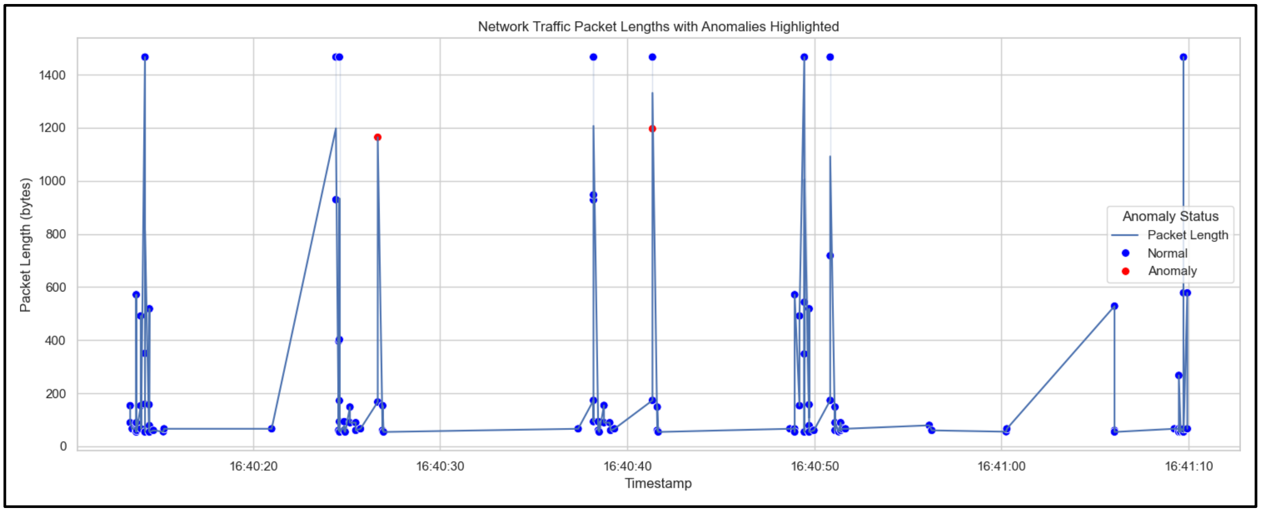


Figure 8: Network traffic packets captured by the tshark

This visual demonstrates the volume and variety of packets captured in the simulation. It verifies the system's ability to handle real-time traffic, including both benign and malicious samples.

5.2. Dataset Preparation

The CICIDS 2017/2018 datasets served as the primary data sources for our ML model training and as such their evaluation. These datasets consist of a wide range of benign and malicious network traffic patterns because they are generated for accurate traffic classification models. The CICIDS datasets were downloaded from their official repository and as such using Python's pandas library all of these CSV files were automatically loaded and unified into a single DataFrame for the subject unified processing. Hence that resulted in a DataFrame which encompassed over 2.5 million rows and 79 columns. The data cleaning and preprocessing phase involved several critical steps:

- Initial exploration showed that the missing values were in specific columns like flow_bytes_s.
- The dataset contained a major chunk of duplicate entries (over 307,000 rows) and as such these duplicate rows were eliminated.
- There were positive infinity values in certain features which were replaced with NaN.
- Columns which showed a single unique value like bwd_psh_flags and bwd_urg_flags were removed to reduce dimensionality.
- The label column was standardized to binary categories i.e. BENIGN and MALICIOUS.
- Continuous features are scaled using StandardScaler.

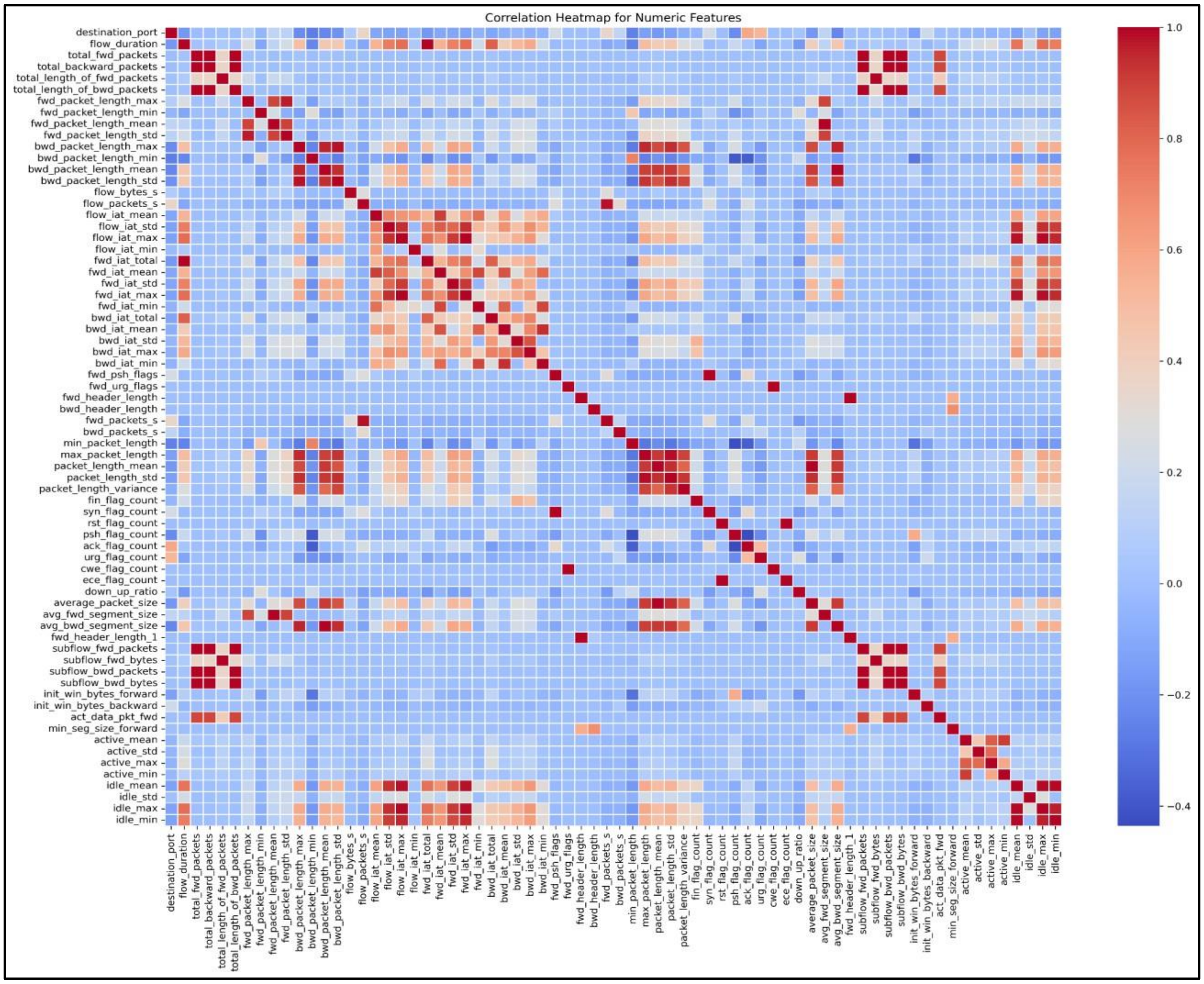


Figure 9: Correlation Heatmap for Numeric Features

The heatmap shows the relationships between numeric features in the dataset. High correlations indicate redundant features, which can be removed to streamline the model and improve its performance. Addressing class imbalance was very important so that the ML models were not biased towards the majority class (benign traffic) and here are the following steps:

- Under-Sampling the Majority Class
- Over-Sampling the Minority Class using SMOTE

5.3. Machine Learning Model Development and Training

The core of the DPI-ML system is in the development and training of such ML models which can detect benign or malicious samples and this phase consists of model selection, training, hyperparameter tuning and evaluation. The three distinct ML models are as follows which were chosen:

- Random Forest
- Support Vector Machine (SVM)
- XGBoost

5.4. Model Training and Hyperparameter Tuning

The training process involved fitting each selected model to the balanced and preprocessed dataset and as such in the post-training each model was tested heavily and then evaluated using the testing dataset on the key metrics i.e. accuracy, precision, recall and F1-score. The confusion matrices were also generated to visualize the models performance in distinguishing between benign and malicious traffic and the results of these evaluation showed that all three models i.e. Random Forest, SVM and XGBoost were great in high classification performance. The ML model XGBoost slightly outperformed the other models in terms of F1-score. The confusion matrices showed that each model has low false positive and false negative rates and as such these findings show the true importance of the integration of ML models with DPI for real-time network traffic analysis and threat detection.

6. Evaluation

This evaluation section presents a complete showcase of the developed Deep Packet Inspection (DPI) system which is combined with Machine Learning (ML) techniques to secure remote work communications and their environments. This evaluation consists of an analysis of classification performance, system efficiency, privacy preservation and benchmarking against existing the ML model solutions which were trained through rigorous testing within this simulated Docker-based remote work environment and as such the DPI-ML system's abilities were checked and tested thoroughly to guarantee the alignment with the research objectives of enhancing the security, optimizing performance and maintaining the said privacy. The core functionality of the DPI-ML system relies on its ability to accurately classify network traffic as benign or malicious in real-time and as such the evaluation of this classification performance was conducted using the testing dataset comprising 340,593 instances which was distributed between benign and malicious traffic. Three ML models i.e. Random Forest, Support Vector Machine (SVM) and XGBoost were checked on their key performance metrics: accuracy, precision, recall and F1-score.

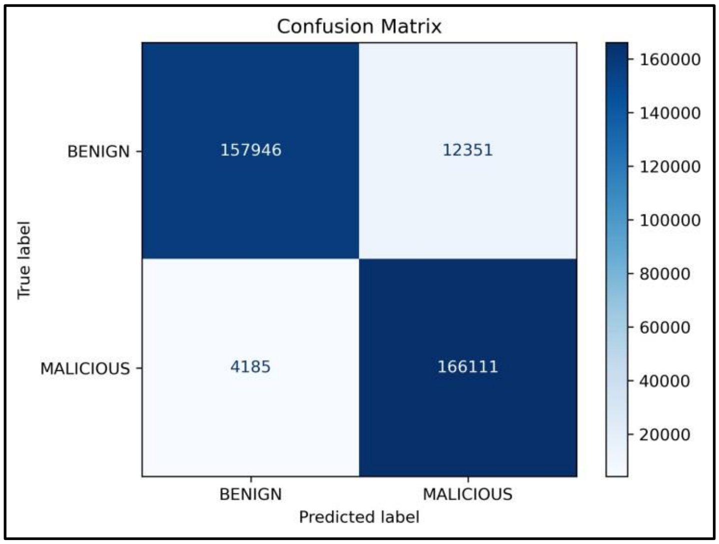


Figure 10: Confusion matrix depicting the results of the SVM and its performance

Random Forest demonstrated exceptional performance by achieving an accuracy of 99.91%, precision of 99.94% for benign traffic and 99.89% for malicious traffic and thus its recall rates were also equally impressive which means that with 99.89% for benign and 99.94% for malicious classes and also with an F1-score of 0.9991 for both classes which shows near-perfect classification results and makes the Random Forest's results look very appealing because in handling large feature sets it is known to be very good in the academic results as well as reducing the overfitting as ensemble learning. Support Vector Machine (SVM) showed very strong performance metrics with the overall accuracy of 95.14%. Precision and recall for benign traffic were 97.42% and 92.75% respectively but the malicious traffic achieved 93.08% precision and 97.54% recall which made its F1-scores of 0.9503 and 0.9526 for benign and malicious classes and made sure that they represented the balanced performance even though it had slightly lower eval metrics than the ensemble models.

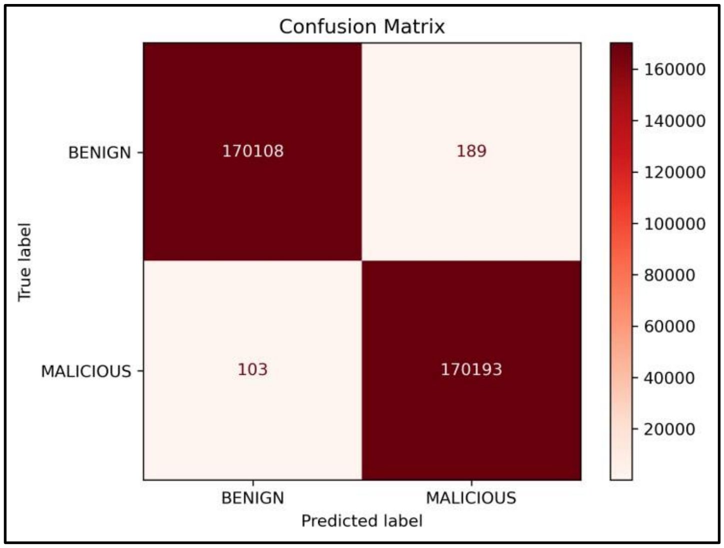


Figure 11: Confusion matrix depicting the results of the Random Forest and its performance

XGBoost was shown to be the top-performing model by achieving an impressive score of accuracy of 99.93% whereas its precision and recall metrics were 99.97% and 99.88% respectively and for benign traffic it was 99.89% and for malicious traffic it was 99.97%. The F1-scores also showed these high performance levels by a whopping 0.9993 for both classes. This is because the XGBoost’s gradient boosting framework is especially known for its high performance and ability to capture complex patterns which in certain cases can even surpass both Random Forest and SVM in classification accuracy and the resultant balance between precision and the recall.

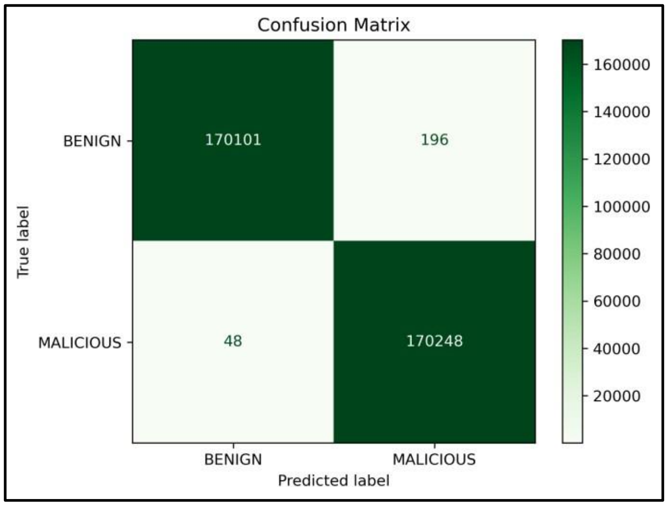


Figure 12: Confusion matrix depicting the results of the XGBoost and its performance

The confusion matrices for all three models showed that minimal misclassifications with negligible false positives and false negatives were achieved and that these results confirm that the DPI-ML system has a high reliability in figuring out the difference between benign and malicious traffic which again cements a powerful threat detection in the remote network activities. Beyond classification accuracy the DPI-ML system's efficiency was tested to be based on latency, throughput, and resource utilization for real-time traffic analysis as discussed above to be in the high-traffic remote work environments and their respective networks. Resource Utilization captured from the tshark tool was monitored to check that the DPI-ML system operates within acceptable resource limitations and thus the CPU usage also is in the relatively low 70% threshold. The memory utilization on the other hand peaked at 60% which was due to the integration of lightweight ML models

7. Conclusion and Future Work

This research has successfully shown that the feasibility and performance of optimizing Deep Packet Inspection (DPI) for securing remote work communications via the integration of subject Machine Learning (ML) techniques is indeed a better approach and has better results. The developed DPI-ML system achieved outstanding classification performance with models such as XGBoost attaining near-perfect accuracy, precision, recall and F1-scores and this was also shown by the system's low latency and high throughput for any real-time threat detection in high-traffic remote work environments. A major contribution of this research study is the incorporation of various privacy-preserving methodologies i.e. Encrypted Traffic Analysis (ETA) which can effectively safeguard the user data and ensure that the regulation standards of the cybersecurity are met with data protection associated with traditional DPI systems. By decentralizing model training and analyzing encrypted metadata only, the DPI-ML framework maintains data confidentiality without sacrificing the accuracy and reliability of threat detection. The key contributions of this research study were:

- The integration of Random Forest, SVM and XGBoost models with DPI significantly improved the accuracy and reliability of network traffic classification, enabling the detection of sophisticated and encrypted threats.
- The DPI-ML system demonstrated exceptional efficiency, achieving low latency and high throughput, essential for real-time threat detection in dynamic remote work environments.
- The implementation of Encrypted Traffic Analysis provides robust privacy protections, ensuring that user data remains confidential and compliant with data protection regulations.
- The system's ability to scale with increasing traffic loads and adapt to diverse network conditions highlights its versatility and potential for deployment across various organizational sizes and structures.

Despite its successes, the DPI-ML system shows certain limitations and its reliance on pre-labeled datasets such as CICIDS 2017/2018 may constrain the system's ability to generalize across entirely novel threat scenarios. Additionally, the computational overhead introduced by privacy-preserving mechanisms, although minimal, could become significant in extremely high-traffic environments or with resource- constrained deployments. By addressing critical challenges related to performance optimization and privacy preservation, the DPI-ML framework offers a robust and scalable solution for safeguarding remote work environments against an evolving array of cyber threats. The promising results achieved in this study lay the groundwork for future advancements and broader adoption of intelligent, privacy-aware network security solutions.

8. References

Ahad, A., Bakar, R.A., Arslan, M. and Ali, M.H., 2023, February. DPIDNS: A Deep Packet Inspection Based IPS for Security Of P4 Network Data Plane. In 2023 International Conference on Smart Computing and Application (ICSCA) (pp. 1-8). IEEE.

Alkhalidi, N.A. and Yaseen, F.A., 2021. FDPHI: Fast Deep Packet Header Inspection for Data Traffic Classification and Management. International Journal of Intelligent Engineering & Systems, 14(4).

Aziz, W.A., Qureshi, H.K., Iqbal, A., Al-Dulaimi, A. and Al–Rubaye, S., 2023, December. Towards Accurate Categorization of Network IP Traffic Using Deep Packet Inspection and Machine Learning. In GLOBECOM 2023-2023 IEEE Global Communications Conference (pp. 01-06). IEEE.

David, J. and Khan, M., Deep Packet Inspection: Leveraging Machine Learning for Efficient Network Security Analysis.

Deri, L. and Fusco, F., 2021, July. Using deep packet inspection in cybertraffic analysis. In 2021 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 89-94). IEEE.

Hussain, S. and Shehzadi, T., 2024. Deep Packet Inspection: Leveraging Machine Learning for Efficient Network Security Analysis. Integrated Journal of Science and Technology, 1(2).

Hypolite, J., Sonchack, J., Hershkop, S., Dautenhahn, N., DeHon, A. and Smith, J.M., 2020, November. DeepMatch: Practical deep packet inspection in the data plane using network processors. In Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies (pp. 336-350).

Jajula, S.K., Tripathi, K. and Bajaj, S.B., 2022. Review of Detection of Packets Inspection and Attacks in Network Security. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2022, Volume 1 (pp. 597-604). Singapore: Springer Nature Singapore.

Khan, F.A. and Ibrahim, A.A., 2024. Machine Learning-based Enhanced Deep Packet Inspection for IP Packet Priority Classification with Differentiated Services Code Point for Advance Network Management. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 16(2), pp.5-12.

Kim, J., Camtepe, S., Baek, J., Susilo, W., Pieprzyk, J. and Nepal, S., 2021, May. P2DPI: practical and privacy-preserving deep packet inspection. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (pp. 135-146).

Liu, Q., Peng, Y., Jiang, H., Wu, J., Wang, T., Peng, T. and Wang, G., 2022. SlimBox: Lightweight packet inspection over encrypted traffic. IEEE Transactions on Dependable and Secure Computing, 20(5), pp.4359-4371.

Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R. and Saberian, M., 2020. Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing, 24(3), pp.1999- 2012.

Lu, Q., Lin, J., Su, Z. and Liu, F., 2021. System design of network data classification based on deep packet inspection. In Journal of Physics: Conference Series (Vol. 1738, No. 1, p. 012118). IOP Publishing.

Malik, A., de Fréin, R., Al-Zeyadi, M. and Andreu-Perez, J., 2020, June. Intelligent SDN traffic classification using deep learning: Deep-SDN. In 2020 2nd International Conference on Computer Communication and the Internet (ICCCI) (pp. 184-189). IEEE.

Ning, J., Huang, X., Poh, G.S., Xu, S., Loh, J.C., Weng, J. and Deng, R.H., 2020. Pine: Enabling privacy- preserving deep packet inspection on TLS with rule-hiding and fast connection establishment. In Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25 (pp. 3-22). Springer International Publishing.

Nkongolo, M., van Deventer, J.P. and Kasongo, S.M., 2022. Using deep packet inspection data to examine subscribers on the network. Procedia Computer Science, 215, pp.182-191.

Nyasore, O.N., Zavarsky, P., Swar, B., Naiyeju, R. and Dabra, S., 2020, May. Deep packet inspection in industrial automation control system to mitigate attacks exploiting modbus/TCP vulnerabilities. In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) (pp. 241-245). IEEE.

Ponomarenko, R.E., Egorov, V.I. and Get'man, A.I., 2023. Challenges in the implementation of systems for deep packet inspection by the method of full protocol decoding. Proceedings of the Institute for System Programming of the RAS, 35(4), pp.45-64.

Sainz, M., Garitano, I., Iturbe, M. and Zurutuza, U., 2020. Deep packet inspection for intelligent intrusion detection in software-defined industrial networks: A proof of concept. Logic Journal of the IGPL, 28(4), pp.461-472.

Sihag, V., Choudhary, G., Vardhan, M., Singh, P. and Seo, J.T., 2021. PICAndro: Packet InspeCtion-Based Android Malware Detection. Security and Communication Networks, 2021(1), p.9099476.

Song, W., Beshley, M., Przystupa, K., Beshley, H., Kochan, O., Pryslupskyi, A., Pieniak, D. and Su, J., 2020. A software deep packet inspection system for network traffic analysis and anomaly detection. Sensors, 20(6), p.1637.

Zhang, X., Geng, W., Song, Y., Cheng, H., Xu, K. and Li, Q., 2023. Privacy-Preserving and Lightweight Verification of Deep Packet Inspection in Clouds. IEEE/ACM Transactions on Networking, 32(1), pp.159- 174.

Çelebi, M., Özbilen, A. and Yavanoğlu, U., 2023. A comprehensive survey on deep packet inspection for advanced network traffic analysis: issues and challenges. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 12(1), pp.1-29.