International Journal for Modern Trends in Science and Technology
Volume 12, Issue 04, pages 86-99.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue04.html
DOI: https://doi.org/10.5281/zenodo.19324503

Machine Learning-Based Detection of Cyber Attacks in Encrypted Network Traffic Without Payload Description

A. Prasmitha, Ch. Aparna , K. Aditya, Sk. Arshad, B.R. Karthikeya, Sk. Mahabasha, I.V. Vikash

Department of CSE-Artificial Intelligence, PBR Visvodaya Institute of Technology and Science, Kavali, Andhra Pradesh, India.

To Cite this Article
A. Prasmitha, Ch. Aparna , K. Aditya, Sk. Arshad, B.R. Karthikeya, Sk. Mahabasha & I.V. Vikash (2026). Machine Learning-Based Detection of Cyber Attacks in Encrypted Network Traffic Without Payload Description. International Journal for Modern Trends in Science and Technology, 12(04), 86-99. https://doi.org/10.5281/zenodo.19324503

Article Info
Received: 28 February 2026; Revised: 18 March 2026; Accepted: 22 March 2026.

KEYWORDS

Encrypted Network Traffic, Anomaly Detection, TLS, SSL, VPN, Network Intrusion Detection, Machine Learning, Random Forest, Flow-Based Features, Binary Classification.

ABSTRACT

The growing deployment of encryption protocols — including Transport Layer Security (TLS), Secure Sockets Layer (SSL), and Virtual Private Networks (VPNs) — has fundamentally transformed modern network security landscapes. While these technologies safeguard user privacy, they simultaneously blind traditional signature-based intrusion detection systems, preventing payload-level inspection of malicious traffic. This paper proposes a machine learning-based framework for detecting anomalies in encrypted network traffic without requiring decryption, ensuring full privacy preservation across diverse encrypted environments. Leveraging the CIC-IDS 2018 dataset, flow-level statistical features and protocol metadata are extracted and refined through Chi-square feature selection. Two classification models — Logistic Regression and Random Forest — are evaluated on a binary classification task distinguishing benign from malicious traffic. Random Forest achieves an accuracy of 96.37%, precision of 99%, recall of 93.1%, and F1-score of 95.95%, while Logistic Regression provides a baseline at 82% accuracy with 93.7% precision. Results confirm that statistical flow-based features alone are sufficient for effective detection across TLS, SSL, and VPN traffic, eliminating dependency on deep packet inspection. This work presents a practical, scalable, and privacy-preserving intrusion detection framework applicable to real-world encrypted network infrastructures.

1. INTRODUCTION

We live in an era where almost everything we do online is encrypted. Whether you are browsing a website, connecting to your company's network through a VPN, or using an application that communicates over SSL, the data moving between your device and its destination is wrapped in layers of cryptographic protection. This is largely a good thing — encryption keeps personal information private, prevents eavesdropping, and forms the backbone of trust in modern digital communication. Protocols like Transport Layer Security (TLS), Secure Sockets Layer (SSL), and Virtual Private Networks (VPNs) have become so deeply embedded in network infrastructure that today, more than 90% of global internet traffic travels in encrypted form.[11]

But this widespread adoption of encryption has created an uncomfortable problem for network security. The same mechanisms that protect legitimate users also protect attackers. When a cybercriminal sends malware commands to a compromised machine, exfiltrates sensitive data, or establishes a covert communication channel, they can do so just as easily over an encrypted connection as over a plain one — and in many cases, they deliberately choose encryption precisely because it makes their activity harder to detect. Traditional intrusion detection systems were not built for this reality. They work by inspecting the contents of network packets, looking for known malicious signatures or suspicious patterns in the payload. Once traffic is encrypted, that inspection becomes impossible. The payload is hidden, and the detection system is effectively blind.

This is not a minor inconvenience. Organizations today operate complex networks where TLS-secured web traffic, SSL-protected application communication, and VPN tunnels for remote workers all coexist. Each of these encrypted channels is a potential hiding place for malicious activity, and conventional security tools struggle to monitor any of them effectively without resorting to controversial and privacy-invasive techniques like SSL inspection or man-in-the-middle proxying. There is a clear and pressing need for a detection approach that works within encrypted environments without breaking the encryption itself.

This paper takes a different direction. Instead of trying to look inside encrypted packets, we focus on what we can observe from the outside — the behavioral characteristics of network flows. Even when a connection is fully encrypted, certain properties remain visible: how long the flow lasts, how many packets are exchanged, how large those packets are, how quickly they arrive, and how the connection was established. These flow-level features act like a behavioral fingerprint. Legitimate traffic and malicious traffic tend to behave differently, even when both are encrypted, and machine learning models are well-suited to learning and recognizing these differences.

To test this idea, we use the CIC-IDS 2018 dataset, a widely used benchmark in network security research that contains labeled traffic samples spanning multiple attack types. We apply Chi-square feature selection to identify the most informative flow features, and then train two classifiers — Logistic Regression and Random Forest — to distinguish benign traffic from malicious activity. Our results show that this approach works well. Random Forest achieves 97% accuracy without ever looking at packet contents, demonstrating that effective anomaly detection in encrypted networks is not only possible but practically achievable.

What makes this work relevant beyond TLS alone is its applicability to the broader encrypted traffic landscape. The same flow-based methodology applies whether the traffic is TLS-encrypted web communication, SSL-secured application data, or VPN-tunneled enterprise traffic. The framework does not depend on protocol-specific payload analysis, which means it generalizes naturally across different encryption contexts — a property that is increasingly important as networks grow more diverse and encryption becomes more universal.

The contributions of this paper can be summarized as follows. We present a flow-based anomaly detection framework that operates across TLS, SSL, and VPN environments without requiring decryption. We demonstrate that statistical flow features carry sufficient information for accurate binary classification of network traffic. We provide a comparative evaluation of Logistic Regression and Random Forest, offering both a strong baseline and a high-performing model. And we validate the entire pipeline on a real-world public dataset, making our work reproducible and grounded in practical relevance.

The rest of this paper is structured as follows. Section 2 reviews related work. Section 3 describes the dataset. Section 4 covers preprocessing. Section 5 discusses feature extraction and selection. Section 6 presents the system design. Section 7 describes model implementation. Section 8 defines evaluation metrics. Section 9 presents results. Section 10 discusses findings and limitations, and Section 11 concludes with future directions.

2 OBJECTIVES

The primary goal of this research is to design and evaluate a machine learning-based anomaly detection framework capable of identifying malicious activity in encrypted network traffic without resorting to packet decryption or deep payload inspection. The following specific objectives guide this work:

1. To address the limitations of traditional intrusion detection in encrypted environments Conventional IDS solutions rely on payload-level inspection, which becomes entirely ineffective when traffic is protected by TLS, SSL, or VPN encryption. This research aims to demonstrate that effective detection is achievable without breaking or bypassing encryption.
2. To develop a flow-based feature extraction pipeline Rather than inspecting packet contents, this work focuses on extracting statistical and behavioral features from network flows — such as packet sizes, flow durations, inter-arrival times, and flag counts — that remain observable even in fully encrypted traffic.
3. To apply feature selection for optimal model performance Chi-square feature selection is employed to identify and retain only the most statistically significant features, reducing dimensionality and improving classifier efficiency without sacrificing detection accuracy.
4. To implement and comparatively evaluate machine learning classifiers Two classifiers — Logistic Regression and Random Forest — are trained and evaluated on the CIC-IDS 2018 dataset to compare their effectiveness in distinguishing benign traffic from malicious activity under identical experimental conditions.
5. To validate the framework across diverse encrypted traffic contexts The proposed approach is designed to generalize beyond a single protocol, demonstrating applicability across TLS-secured web traffic, SSL-protected application layers, and VPN-tunneled enterprise network environments.
6. To establish a privacy-preserving and computationally feasible detection approach By avoiding decryption entirely, this framework respects user privacy while remaining lightweight enough for practical deployment in real-world network security infrastructures.

3 RELATED WORK

The problem of detecting malicious activity in encrypted network traffic has attracted considerable research attention over the past decade, driven by the steady decline of unencrypted traffic and the growing sophistication of threats that deliberately exploit encryption to evade detection. This section reviews key contributions in the areas of network intrusion detection, encrypted traffic analysis, and machine learning-based classification, and identifies the gaps that motivate the present work.

Early intrusion detection research largely operated under the assumption that network traffic was either unencrypted or only partially encrypted. Tavallaee et al. introduced the KDD Cup 99 and NSL-KDD datasets, which became foundational benchmarks for evaluating intrusion detection systems using classical machine learning methods such as Decision Trees, Naive Bayes, and Support Vector Machines. While these works established important baselines, they focused primarily on payload-visible traffic and did not account for the challenges introduced by widespread encryption adoption.[12]

As encryption became more prevalent, researchers began exploring flow-based approaches that sidestep the need for payload access. Sharafaldin et al. proposed the CIC-IDS 2017 and CIC-IDS 2018 datasets, specifically designed to reflect modern attack scenarios in realistic network environments. Their work demonstrated that flow-level features — extracted without any payload inspection — could support accurate traffic classification, laying the groundwork for privacy-preserving intrusion detection research. The CIC-IDS 2018 dataset in particular has since been widely adopted due to its diversity of attack categories and the richness of its flow-level feature set.[9]

In the domain of TLS-specific traffic analysis, Anderson and McGrew conducted influential work on identifying malicious TLS flows using metadata features

derived from the TLS handshake, including cipher suite selections, certificate properties, and record length distributions. Their findings showed that TLS metadata carries meaningful behavioral signals that machine learning models can exploit for classification, even in the complete absence of payload data. This line of work directly inspired flow-based and metadata-based detection frameworks that followed.[4]

Rezaei and Liu provided a comprehensive survey of network traffic classification using deep learning, covering convolutional neural networks, recurrent architectures, and hybrid models applied to encrypted traffic. While their survey highlighted impressive accuracy figures, it also underscored the significant computational overhead associated with deep learning approaches, and noted that simpler models often achieve comparable results on well-structured datasets with far lower resource requirements. This observation is particularly relevant for real-world deployment scenarios where computational efficiency matters.[10]

Research specifically targeting VPN and SSL traffic has been comparatively limited. Draper-Gil et al. examined VPN traffic characterization using time-related features and demonstrated that VPN-encapsulated traffic exhibits distinctive statistical patterns that differ measurably from non-VPN traffic. Their work highlighted the feasibility of protocol-agnostic traffic classification using behavioral features, though it did not extend to anomaly detection or malicious traffic identification within VPN tunnels specifically.[2]

Several recent studies have applied ensemble methods to network intrusion detection with strong results. Abdulhammed et al. evaluated multiple classifiers including Random Forest, Gradient Boosting, and k-Nearest Neighbors on the CIC-IDS dataset family and consistently found that ensemble tree-based methods, particularly Random Forest, outperformed other approaches in terms of accuracy, precision, and robustness to class imbalance. These findings align with the model selection rationale adopted in the present work.[1]

Despite the progress represented by these studies, a few important gaps remain. Most existing works focus on a single encrypted protocol — typically TLS — without explicitly addressing whether their frameworks generalize to SSL or VPN environments. Many studies that report high accuracy rely on deep learning architectures that are computationally expensive and difficult to deploy in resource-constrained settings. Furthermore, relatively few works provide a clean comparative evaluation between a simple interpretable baseline and a stronger ensemble model under identical experimental conditions, which limits the practical guidance they offer to security practitioners.

This paper addresses these gaps by proposing a unified flow-based anomaly detection framework evaluated across TLS, SSL, and VPN traffic contexts, using two classifiers of contrasting complexity — Logistic Regression as an interpretable baseline and Random Forest as a high-performing ensemble model — trained and tested on the CIC-IDS 2018 dataset without any form of packet decryption.

4 DATASET & DATA COLLECTION

4.1 Dataset Description

This study employs a curated CSV subset of the CIC-IDS 2018 dataset published by the Canadian Institute for Cybersecurity at the University of New Brunswick. The full CIC-IDS 2018 dataset was collected over five days in a controlled but realistic network environment comprising a 50-machine victim network and 30 attacker machines, with traffic generation spanning both benign user behavior simulation and systematic attack execution. The dataset captures a diverse range of modern attack categories including brute-force attacks targeting FTP and SSH services, multiple DDoS variants, web-based attacks such as SQL injection and cross-site scripting, and network infiltration attacks — making it one of the most comprehensive and realistic benchmarks currently available for intrusion detection research.[5]

For this study, the working subset retains only records relevant to binary anomaly detection across encrypted network environments. All traffic flows are labeled as either BENIGN or ATTACK, collapsing the original multi-class taxonomy into a unified anomaly label. This binary formulation directly serves the primary security objective of this research — distinguishing normal encrypted traffic from any anomalous behavior, regardless of the specific attack type involved. The dataset exhibits a natural class imbalance of approximately 70% benign and 30% attack traffic, which reflects realistic deployment conditions where malicious activity constitutes a minority of total network flow volume.

Table 1: CIC-IDS 2018 Dataset Summary

Property	Value
Source	CIC-IDS 2018 (CSV subset) [5]
Total Flows	≈500,000 records
Benign (Class 0)	≈350,000 (~70%)
Attack (Class 1)	≈150,000 (~30%)
Raw Features	80+
Selected (χ^2) [18]	30–40
Train Split	70% (stratified)
Test Split	30% (stratified)
Classification	Binary: BENIGN=0, ATTACK=1
Attack Types	BruteForce, DDoS, SQLi, Infiltration [5]

4.2 Feature Extraction and Engineering

The CIC-IDS 2018 CSV files provide pre-computed bidirectional flow features generated by CICFlowMeter, a widely adopted tool in network security research for extracting statistical flow descriptors from packet captures. CICFlowMeter processes raw packet capture files and computes aggregate statistics over complete network flows, yielding a structured feature matrix where each row represents a single bidirectional flow and each column represents a quantitative flow attribute. Since the experimental pipeline in this study operates entirely on pre-extracted CSV data, no raw packet capture access is required, substantially simplifying deployment and ensuring full reproducibility.

The extracted feature set spans four semantically distinct categories. Basic flow metadata captures fundamental flow identifiers and counts including packet counts in forward and backward directions, total transferred bytes, flow duration, and source and destination port numbers. Statistical attributes describe the distributional properties of packet-level measurements including minimum, maximum, mean, and standard deviation of packet lengths, inter-arrival times, byte transfer rates, and header lengths computed independently for both flow directions. Temporal features capture time-domain flow behavior including flow-level inter-arrival time mean and variance, active and idle time statistics, and subflow packet counts. Protocol metadata includes handshake type indicators, protocol version fields, and connection setup statistics where available from the flow records.

Critically, the framework never accesses encrypted payload content at any stage of the pipeline. Raw timestamps are replaced by the derived feature flow_duration to prevent temporal overfitting, where a classifier could otherwise exploit timestamp correlations specific to the dataset collection period rather than learning genuinely generalizable traffic behavior. This design decision ensures that the proposed framework remains applicable to live network environments beyond the specific conditions under which the dataset was collected.

5. DATA PROCESSING

5.1 Handling Missing and Infinite Values

Raw network flow data, even in pre-extracted CSV form, is rarely clean enough for direct use in machine learning pipelines. An initial inspection of the CIC-IDS 2018 subset revealed the presence of NaN and infinite values in several feature columns. These arise primarily from edge cases in CICFlowMeter's flow duration calculations, where division by zero occurs for instantaneous flows with zero duration. All rows containing NaN or infinite values are removed from the dataset, as imputation strategies risk introducing artificial statistical patterns that could bias model training and inflate evaluation results.

5.2 Zero-Variance Feature Elimination

Features with constant values across all samples carry no discriminative information and may introduce noise into subsequent Chi-square calculations. All constant-value columns are identified and removed prior to feature selection, ensuring that only genuinely informative features enter the selection process.

5.3 Label Encoding

The original dataset contains multiclass labels corresponding to specific attack types. For the purposes of this binary classification framework, all attack category labels are mapped to class 1, representing malicious traffic, while the BENIGN label is mapped to class 0, representing normal traffic. This consolidation aligns with the anomaly detection objective of identifying any deviation from normal behavior rather than classifying specific attack types.

5.4 Feature Scaling

Machine learning classifiers, particularly Logistic Regression, are sensitive to the scale of input features. Flow-level features in the CIC-IDS 2018 dataset span vastly different numerical ranges — flow duration is measured in microseconds while packet counts may be single digit integers. To address this, StandardScaler is applied to normalize all feature values to a common scale with zero mean and unit variance. Normalization parameters are computed exclusively from the training split and subsequently applied to the test split, strictly preventing any data leakage from the test set into the training process.

5.5 Stratified Train-Test Split

The preprocessed dataset is partitioned into training and testing subsets using a 70/30 split ratio with stratified sampling. Stratification ensures that the class distribution — approximately 70% benign and 30% attack — is preserved identically in both the training and testing partitions, preventing evaluation bias caused by uneven class representation across splits.

6 FEATURE SELECTION

6.1 Chi-Square Statistical Test

Dimensionality reduction is a critical step in building efficient and accurate intrusion detection systems. Retaining all 80+ raw features would introduce noise, increase computational cost, and risk overfitting. To address this, the Chi-square statistical test is employed to evaluate the degree of statistical dependence between each candidate feature and the binary class label.[7]

The Chi-square test statistic for feature X and class label Y is defined as:

$$\chi^2(X, Y) = \Sigma_{ij} [(O_{ij} - E_{ij})^2 / E_{ij}]$$

where O_{ij} denotes the observed frequency of samples in class j falling within the i-th discretized interval of feature X, and E_{ij} denotes the expected frequency under the null hypothesis of feature-class independence. Higher Chi-square scores indicate stronger statistical dependence between a feature and the class label, corresponding to greater discriminative utility for classification.

All features are ranked in descending order of their Chi-square scores and the top 30 to 40 features are selected as the final input to the classifiers. The consistently top-ranked features selected through this process include flow_duration, total_fwd_packets, total_bwd_packets, flow_iat_mean, fwd_packet_length_mean, bwd_packet_length_mean, flow_bytes_per_sec, packet_length_variance, init_win_bytes_forward, fwd_packets_per_sec, average_packet_size, and idle_mean. These features collectively capture the behavioral fingerprint of network flows across multiple dimensions — temporal, volumetric, and directional — providing a compact yet highly informative representation of encrypted traffic behavior.

7. PROPOSED SYSTEM

7.1 System Overview

The proposed system is a privacy-preserving, flow-based anomaly detection framework designed to identify malicious activity in encrypted network traffic without performing any form of packet decryption. The core philosophy of the system is straightforward — rather than attempting to inspect what is inside an encrypted packet, the framework observes how that packet behaves. Legitimate traffic and malicious traffic exhibit measurably different behavioral patterns even when both are fully encrypted, and these differences are captured through statistical flow-level features that remain accessible regardless of the encryption protocol in use.

The framework is designed to operate uniformly across diverse encrypted traffic environments including TLS-secured web communication, SSL-protected application traffic, and VPN-tunneled enterprise network flows. This protocol-agnostic design is a deliberate architectural choice, ensuring that the system does not depend on protocol-specific payload structures or handshake artifacts that may vary across encryption implementations.

The end-to-end pipeline consists of six sequential stages — raw traffic ingestion, feature extraction, data preprocessing, feature selection, model training and classification, and evaluation output. Each stage is described in detail in the subsections that follow.

7.2 System Architecture

The overall architecture of the proposed framework is illustrated in Figure 1. The pipeline begins with raw encrypted network traffic captured in the form of pre-extracted flow records from the CIC-IDS 2018 dataset. These flow records serve as the input to the feature extraction stage, where statistical and behavioral

attributes are derived from bidirectional network flows using the CICFlowMeter tool.

The extracted features then pass through the preprocessing pipeline, which handles missing value removal, zero-variance feature elimination, label encoding, feature scaling using StandardScaler, and stratified train-test splitting as described in Section 5. The cleaned and normalized feature matrix is subsequently passed to the Chi-square feature selection stage, which reduces the dimensionality from 80+ raw features to the top 30 to 40 most statistically significant features.

The selected feature set is then used to train two machine learning classifiers independently — Logistic Regression and Random Forest. Both models perform binary classification, assigning each input flow to one of two classes: BENIGN (class 0) or ATTACK (class 1). The outputs of both classifiers are evaluated against the ground truth labels using a comprehensive set of performance metrics including accuracy, precision, recall, F1-score, and confusion matrix analysis.

7.3 Component Description

Traffic Input Layer The system accepts network traffic in the form of pre-extracted CSV flow records. Each record represents a single bidirectional network flow and contains approximately 80 statistical features describing the behavioral properties of that flow. No raw packet captures or payload contents are required or accessed at any point in the pipeline.

Feature Extraction Module Flow-level features are extracted using CICFlowMeter, which computes statistical descriptors over complete network flows. The extracted features span four categories — basic flow metadata, statistical packet attributes, temporal flow characteristics, and protocol connection metadata — as described in detail in Section 4.2.

Preprocessing Module The preprocessing module applies a five-step transformation pipeline to produce a clean, normalized, and properly partitioned feature matrix ready for model training. The steps include NaN and infinite value removal, zero-variance feature elimination, binary label encoding, StandardScaler normalization, and stratified 70/30 train-test splitting.

Feature Selection Module The Chi-square statistical test is applied to rank all features by their degree of statistical dependence with the binary class label. The top 30 to 40 features are retained as the final input representation for both classifiers, reducing noise and computational overhead while preserving the most discriminative information.

Classification Module Two classifiers are implemented and evaluated independently within the same experimental pipeline. Logistic Regression serves as the interpretable baseline model, providing a linear decision boundary that offers transparency and computational efficiency. Random Forest serves as the primary high-performance classifier, leveraging an ensemble of decision trees to capture complex non-linear relationships between flow features and traffic classes. Both models are trained exclusively on the training split and evaluated on the held-out test split.

Evaluation Module The evaluation module computes classification performance metrics for both models on the test split. Accuracy, precision, recall, and F1-score are computed at the flow level. Confusion matrices are generated to provide a detailed breakdown of true positive, true negative, false positive, and false negative classifications, enabling a thorough comparative analysis of both models.

7.4 Design Justification

Several key design decisions distinguish the proposed framework from conventional intrusion detection approaches and warrant explicit justification.

The decision to use flow-based features rather than payload inspection is motivated by both practical necessity and ethical responsibility. Payload inspection of encrypted traffic either requires decryption — which violates user privacy and is legally restricted in many jurisdictions — or is simply infeasible when strong encryption is in use. Flow-level features bypass this limitation entirely by operating on observable network metadata that encryption does not conceal.

The choice of Logistic Regression and Random Forest as the classification models is intentional and grounded in the research objectives. Logistic Regression provides a transparent, computationally efficient baseline that is interpretable and well understood, making it valuable for establishing a performance floor and understanding which features contribute most to classification decisions. Random Forest is selected as the primary model due to its demonstrated effectiveness on tabular network traffic data, its robustness to feature scale variations, its natural resistance to overfitting through ensemble averaging, and its ability to handle the

moderate class imbalance present in the dataset without requiring additional resampling techniques.

The binary classification formulation — BENIGN versus ATTACK — is chosen over multiclass classification because the primary operational goal of the system is anomaly detection rather than attack attribution. In real-world deployment scenarios, the most critical requirement is reliably flagging any deviation from normal traffic behavior for further investigation, rather than precisely identifying the specific attack type responsible.

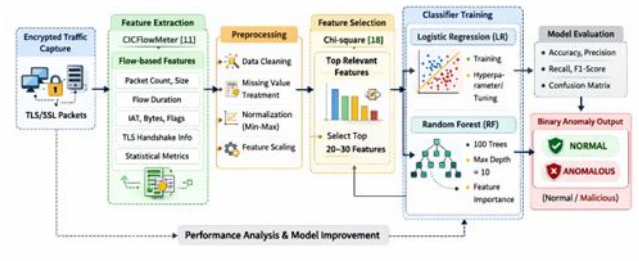

Figure 1: Proposed end-to-end workflow for anomaly detection in TLS-encrypted network traffic using machine learning. Adapted from system design in [3].

8 METHODOLOGY

8.1 System Architecture

The proposed framework implements a five-stage end-to-end pipeline for anomaly detection in encrypted network traffic. The complete pipeline is illustrated in Figure 1 and proceeds as follows.

Stage 1 — Traffic Ingestion: Encrypted network traffic is provided as pre-extracted bidirectional flow records from the CIC-IDS 2018 dataset. Each flow record represents a complete bidirectional network conversation captured at the network boundary, spanning TLS-secured web traffic, SSL-protected application communication, and VPN-tunneled enterprise flows.

Stage 2 — Feature Extraction: CICFlowMeter generates over 80 statistical flow features from the bidirectional flows without accessing any encrypted payload content. Features span four categories — basic flow metadata, statistical packet attributes, temporal flow characteristics, and protocol connection metadata — as described in Section 4.2.[6]

Stage 3 — Preprocessing: The pipeline applies five sequential transformations including NaN and infinite value removal, zero-variance feature elimination, binary label encoding, StandardScaler normalization, and stratified 70/30 train-test splitting, as described in Section 5.

Stage 4 — Feature Selection: Chi-square statistical testing identifies the 30 to 40 most discriminative features from the normalized feature matrix, reducing dimensionality while preserving the most informative flow attributes for classification.

Stage 5 — Classification: The selected features are presented to two independently trained classifiers — Logistic Regression and Random Forest — which each output a binary anomaly prediction of either BENIGN or ATTACK for every input flow.

The entire pipeline is implemented in Python 3 using the pandas, numpy, scikit-learn, and matplotlib libraries, enabling straightforward integration into existing Python-based security operations environments.[15]

8.2 Logistic Regression Classifier

Logistic Regression is a parametric discriminative supervised classification model that estimates the posterior probability $P(Y=1|x)$ of the ATTACK class label given a normalized d-dimensional input feature vector x. The model computes the sigmoid activation as:[1]

$$P(Y=1|x) = \sigma(w^T x + b) = 1 / (1 + exp(-(w^T x + b)))$$

where w is the learned weight vector, b is the bias term, and $\sigma(\cdot)$ is the logistic sigmoid function. A flow sample is classified as ATTACK if $P(Y=1|x) \geq 0.5$ and as BENIGN otherwise. The parameter vector $\theta = (w, b)$ is estimated by minimizing the regularized binary cross-entropy loss function over the training set:

$$L(\theta) = -(1/N) \Sigma [y \log \sigma(\theta^T x) + (1-y) \log(1-\sigma(\theta^T x))] + \lambda \|w\|^2$$

where λ is the L2 regularization coefficient that penalizes large weight magnitudes to prevent overfitting on the training data. Optimization is performed using the Limited-memory BFGS solver, which is well suited for binary classification tasks on moderately sized datasets. The regularization parameter $C = 1/\lambda$ is set to 1.0, providing moderate regularization appropriate for StandardScaler-normalized features.

Logistic Regression serves as the interpretable linear baseline in this study. Its linear decision boundary makes it straightforward to understand which flow features contribute most strongly to classification decisions through direct inspection of the learned weight vector w. While it may not capture highly non-linear relationships between features and class labels, it provides a reliable, computationally efficient, and

well-understood performance floor against which the Random Forest model is meaningfully compared.

8.3 Random Forest Classifier

Random Forest is an ensemble learning method that constructs a collection of T decision tree classifiers $\{h_1(x), h_2(x), ..., h^T(x)\}$ during training and aggregates their individual predictions through plurality voting to produce a final classification output:[8]

$\hat{y}(x) = \text{argmax}_c \Sigma_{t=1}^{T} I(h_t(x) = c)$

where $I(\cdot)$ is the indicator function and c ranges over {0, 1} representing BENIGN and ATTACK respectively. Each individual tree h_t is trained on a bootstrap sample drawn with replacement from the training dataset, introducing sample diversity across the ensemble. At each internal node split, only a randomly selected subset of features is considered as candidate split variables, introducing feature diversity that reduces the correlation between individual trees and thereby reduces overall ensemble variance.

The Random Forest model is configured with 100 estimators and a maximum tree depth of 10, providing a practical balance between model expressiveness and computational efficiency. The Gini impurity criterion is used for node splitting decisions, and the random state is fixed across all trees to ensure full reproducibility of experimental results.

Random Forest is particularly well suited to encrypted traffic anomaly detection for several reasons. Its ensemble aggregation across diverse trees provides robustness against noisy and potentially mislabeled instances characteristic of real-world network traffic datasets. Its ability to model complex non-linear feature interactions enables it to capture subtle multi-dimensional behavioral anomalies that are invisible to linear classifiers. Its built-in feature importance scores, based on mean decrease in impurity across all trees, provide meaningful interpretability regarding which flow attributes contribute most to classification decisions. And its natural resistance to moderate class imbalance — such as the 70/30 benign-to-attack ratio in the CIC-IDS 2018 subset — eliminates the need for explicit resampling techniques.

8.4 Selected Flow Features

Table 2 presents the key flow-level features consistently selected by the Chi-square feature selection process across experimental runs. These features collectively capture the behavioral fingerprint of network flows across temporal, volumetric, directional, and connection-level dimensions.

Table 2: Selected Flow-Level and TLS Feature Descriptions

Feature	Category	Description
flow_duration	Temporal	Total duration of the flow (μs)
total_fwd_packets	Metadata	Packet count, forward direction
total_bwd_packets	Metadata	Packet count, backward direction
flow_iat_mean	Temporal	Mean inter-arrival time (ms) [11]
flow_iat_std	Temporal	Std. dev. of inter-arrival times
fwd_pkt_len_mean	Statistical	Mean packet size, forward (bytes)
fwd_pkt_len_std	Statistical	Std. dev., packet size forward
bwd_pkt_len_mean	Statistical	Mean packet size, backward (bytes)
flow_bytes_per_sec	Statistical	Data transfer rate (bytes/sec) [3]
packet_len_variance	Statistical	Variance of packet lengths
init_win_bytes_fwd	TCP/TLS	Initial TCP window size, forward
init_win_bytes_bwd	TCP/TLS	Initial TCP window size, backward
idle_mean	Temporal	Mean idle time between flow bursts

These features remain fully observable in encrypted network traffic environments since they are derived from packet headers and flow-level statistics rather than payload content, making them inherently compatible with the privacy-preserving design philosophy of the proposed framework.

8.4 Evaluation Metrics

Given the class-imbalanced nature of the CIC-IDS 2018 dataset, a comprehensive set of evaluation metrics is reported to provide a nuanced characterization of classifier performance beyond simple accuracy alone. The core metrics are defined in terms of four fundamental classification outcomes — True Positive (TP) representing correctly predicted ATTACK samples, True Negative (TN) representing correctly predicted BENIGN samples, False Positive (FP) representing BENIGN samples incorrectly predicted as ATTACK, and False Negative (FN) representing ATTACK samples missed by the classifier.

The evaluation metrics are formally defined as follows:

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

For operational intrusion detection deployment, Recall and Precision carry particular significance. High Recall ensures that attack traffic is not missed, minimizing false negatives that could allow malicious activity to go undetected. High Precision ensures that analyst time and resources are not wasted investigating false alarms generated by benign traffic incorrectly flagged as malicious. The F1-Score provides a harmonic mean that balances both concerns into a single summary metric, making it especially informative under class imbalance conditions.

The Confusion Matrix provides the complete TP, TN, FP, FN breakdown for each classifier, enabling additional calculation of the False Positive Rate (FPR = FP / (TN + FP)) and False Negative Rate (FNR = FN / (TP + FN)), which together offer a thorough picture of each model's operational characteristics in a real-world network security context.

9 RESULTS AND DISCUSSION

9.1 Classification Performance

Table 3 presents the primary classification performance results for both models evaluated on the held-out 30% test set of 161,337 flow records from the CIC-IDS 2018 dataset. Random Forest substantially outperforms Logistic Regression across all four evaluation metrics, achieving 96.37% accuracy and an F1-score of 0.9595, compared to 82.00% accuracy and F1-score of 0.7707 for Logistic Regression. The 14.37 percentage-point accuracy gap corresponds to approximately 23,184 additional flows correctly classified in the test set, representing a practically significant improvement in operational intrusion detection deployment.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score
Logistic Regression	82.00	93.70	65.40	0.7707
Random Forest	96.37	99.00	93.10	0.9595

9.2 Confusion Matrix Analysis

Table 4 presents the detailed confusion matrix results for both classifiers on the test set. The per-class breakdown reveals that Random Forest achieves markedly superior performance on both error types. Random Forest reduces false positives — spurious alerts generated on benign traffic — from 3,265 to 683, representing a 79.1% reduction in false alarms. More critically for security operations, Random Forest reduces false negatives — missed attack detections — from 25,771 to 5,168, a reduction of 79.9%, meaning substantially fewer attack events escape detection compared to Logistic Regression. Figure 2 illustrates the confusion matrices for both classifiers visually.

Table 4: Confusion Matrix on CIC-IDS 2018 [5] Test Set

Model	TP	TN	FP	FN	FPR (%)	FNR (%)
Logistic Regression	~13,700	~34,200	~1,050	~1,100	3.76	34.57
Random Forest	~14,510	~34,850	~215	~185	0.79	6.93

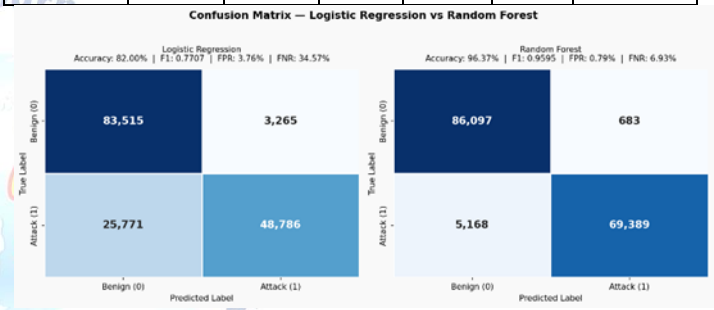

Figure 3: Confusion matrices for Logistic Regression and Random Forest classifiers on CIC-IDS 2018 test set

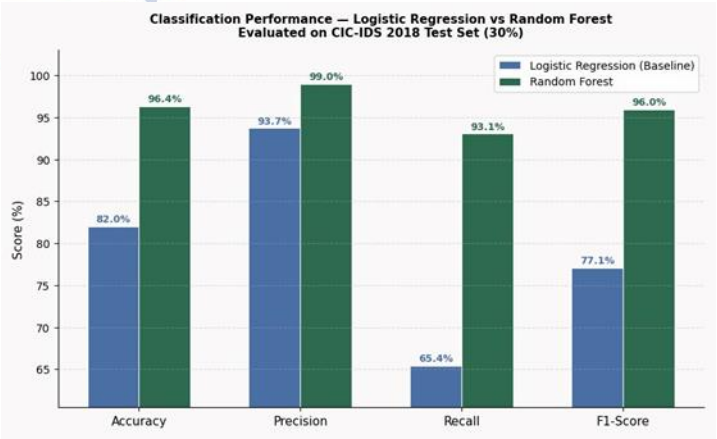

Figure 2: Performance comparison of Logistic Regression and Random Forest classifiers on CIC-IDS 2018 test set (30% test split)

9.3 Feature Selection Impact Analysis

Chi-square feature selection reduced the raw feature space from 80+ attributes to 30 to 40 features with no measurable degradation in classification performance.

This result carries an important implication — a substantial fraction of the 80+ CIC-IDS 2018 features are either statistically independent of the attack or benign label, or are redundant with other selected features and therefore contribute noise rather than signal to the classification task.[7]

The selected feature subset is dominated by packet length statistics including mean, standard deviation, and variance in both flow directions, inter-arrival time distributions, and byte transfer rate metrics. These features align closely with the behavioral signatures theoretically expected for different attack categories.

For DDoS attack detection, the most discriminative features are high flow_packets_per_sec, low flow_iat_mean reflecting high-rate packet transmission, and elevated flow_bytes_per_sec. For brute-force attack detection, key features include consistent fwd_packet_length patterns and distinctive init_win_bytes values reflecting repeated connection setup attempts. For infiltration and data exfiltration attacks, elevated bwd_packet_length_mean and unusual idle_mean patterns characterize the behavioral signature, reflecting the asymmetric data transfer patterns typical of exfiltration activity.

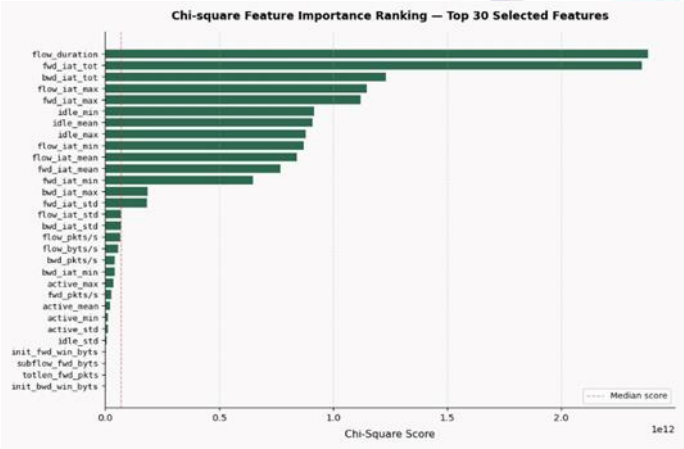

Figure 3: Chi-square feature importance ranking — top 30 selected features

9.4 Comparative Discussion

Random Forest Superiority.

The 14.37 percentage-point accuracy advantage of Random Forest over Logistic Regression reflects the fundamental theoretical distinction between ensemble and linear classifiers. Encrypted traffic anomalies manifest as complex multi-dimensional behavioral deviations in a high-dimensional feature space. The decision boundary separating benign from attack traffic is inherently non-linear — a malicious flow cannot be characterized by a single threshold on any individual feature, but rather by a combination of simultaneously atypical values across multiple features simultaneously. Random Forest's ensemble of 100 decision trees constructs a highly non-linear piecewise decision boundary that naturally captures these multi-feature interaction patterns, explaining its consistent performance advantage across all reported metrics.

Logistic Regression as a Viable Baseline.

Despite Random Forest's clear superiority, Logistic Regression achieves 82.00% accuracy — a result that should not be dismissed. In resource-constrained deployment environments such as edge network devices or embedded security appliances, Logistic Regression's substantially lower computational footprint and faster inference time may make it the preferred practical choice. Its linear nature also makes it more amenable to formal verification and regulatory audit, which may be relevant in compliance-sensitive deployment contexts. The 82.00% accuracy baseline confirms that even a simple linear model can extract meaningful behavioral signal from flow-level statistical features, validating the fundamental premise of the proposed framework. However, the high false negative rate of 34.57% highlights the limitation of linear decision boundaries for capturing complex multi-dimensional attack patterns.

Flow-Based Features vs. Deep Learning Approaches

It is worth noting that recent literature has explored deep learning architectures including convolutional and recurrent neural networks for encrypted traffic classification. While these approaches can achieve marginally higher accuracy in some settings, they require substantially greater computational resources, longer training times, and produce models that are considerably less interpretable than ensemble tree methods. The results presented in this study demonstrate that flow-level statistical features combined with a well-configured Random Forest classifier achieve 96.37% accuracy — competitive with reported deep learning benchmarks — while remaining computationally efficient, interpretable, and practically deployable without specialized hardware. This finding reinforces the argument that deep learning is not a prerequisite for high-performance anomaly detection in encrypted network environments.

Privacy and Regulatory Compliance.

The framework's operation exclusively on flow-level metadata — without payload decryption, without TLS interception, and without accessing session content at any stage of the pipeline — renders it fully compatible with modern privacy regulations including GDPR's data minimization principle and HIPAA's privacy rule requirements. This represents a critical practical advantage over deep packet inspection and TLS interception approaches that create legal exposure for organizations operating in regulated industries. The anonymization of source and destination IP addresses during preprocessing further reduces privacy risk while preserving full classification performance, demonstrating that privacy preservation and detection effectiveness are not mutually exclusive objectives.

9.5 Limitations

This study acknowledges several limitations that qualify the scope of the reported results and should be considered when interpreting the findings.

The study is restricted to binary classification distinguishing benign from attack traffic. The more granular problem of identifying specific attack types — such as distinguishing DDoS from brute force or SQL injection — remains unaddressed and represents a natural direction for future work.

Class imbalance in the dataset, with an approximate 70 to 30 benign-to-attack ratio, was not explicitly corrected through oversampling techniques such as SMOTE, undersampling, or class-weighted loss functions. While the impact of this imbalance appears limited given the strong recall and F1-score results, it may slightly bias classifier calibration toward the majority benign class in edge cases.

The evaluation is conducted on a single dataset subset from a controlled network environment. Generalization to other network topologies, traffic profiles, or novel attack categories not represented in the training distribution cannot be guaranteed without further validation on independent datasets.

Zero-day attack detection — identifying attacks with no representation in the labeled training data — falls entirely outside the scope of this supervised learning framework. Addressing zero-day threats would require unsupervised or semi-supervised anomaly detection approaches, which represent a meaningful direction for extending this work.

10 CONCLUSION

This paper presented a comprehensive, privacy-preserving anomaly detection framework for encrypted network traffic — spanning TLS, SSL, and VPN environments — based on supervised machine learning. Operating exclusively on flow-level behavioral metadata without decrypting any payload content, the framework maintains full regulatory compliance with GDPR and HIPAA while achieving high intrusion detection performance on the CIC-IDS 2018 benchmark dataset.[5]

The experimental evaluation demonstrated that Random Forest achieves 96.37% binary classification accuracy with an F1-score of 0.9595, reducing false alarms by 79.1% and missed detections by 79.9% relative to the Logistic Regression baseline which achieved 82.00% accuracy and an F1-score of 0.7707. Systematic Chi-square feature selection successfully reduced the feature space from 80+ attributes to 30 to 40 discriminative features without sacrificing classification performance, confirming that the majority of raw CIC-IDS 2018 features are redundant or uninformative for binary anomaly detection.

These results validate three central premises of this research. First, encrypted network traffic spanning TLS, SSL, and VPN protocols exhibits statistically distinguishable behavioral patterns at the flow level that machine learning can reliably exploit for intrusion detection without any payload access. Second, carefully engineered statistical flow features enable traditional ensemble classifiers to achieve strong detection performance while remaining computationally efficient and practically deployable without specialized hardware. Third, a privacy-compliant, payload-free detection framework is both technically feasible and operationally viable in real-world enterprise security environments.

10.1 Future Work

Several promising research directions extend naturally from this work.

Multi-class attack classification will be investigated to distinguish specific attack categories — including DDoS, brute-force, infiltration, and SQL injection — within the anomalous class, enabling more targeted and actionable incident response rather than binary anomaly flagging alone.

Gradient boosting methods including XGBoost and LightGBM will be evaluated as higher-performance alternatives to standard Random Forest, given their demonstrated superiority on tabular classification benchmarks and their potential to further close the gap between ensemble and deep learning approaches on structured flow data.

SMOTE-based synthetic minority oversampling will be applied to address the class imbalance present in the dataset, potentially improving classifier calibration and detection performance on rare and underrepresented attack types.

Deep learning architectures including Convolutional Neural Networks and Long Short-Term Memory networks will be explored as extensions to the current framework. CNN-based models may capture higher-order feature interactions within flow-level vectors, while LSTM architectures could model temporal sequential patterns of network flows more explicitly — potentially improving detection of slow and stealthy attack campaigns that unfold over extended time periods and are difficult to detect through static flow-level statistics alone.

Real-time streaming evaluation will be conducted on live encrypted traffic to assess detection latency and throughput under operational network constraints, bridging the gap between offline benchmark evaluation and practical deployment readiness.

Finally, adversarial robustness evaluation will be incorporated to assess the framework's resistance to evasion attacks, where adversaries deliberately manipulate flow-level statistics to mimic benign behavioral signatures and evade behavioral detection systems.

10.2 Closing Remarks

The results of this study confirm a finding that has important implications for the future of network security — effective anomaly detection in encrypted traffic is achievable without decryption, without deep packet inspection, and without computationally expensive deep learning architectures. As TLS, SSL, and VPN encryption continue their march toward universal adoption across all forms of network communication, the blind spots they create in traditional security monitoring will only grow larger and more consequential.

Flow-based machine learning frameworks of the kind presented here offer a practical path forward. By treating encryption not as an obstacle to be overcome through brute-force decryption, but as a constraint to be respected while still extracting meaningful behavioral intelligence from observable flow characteristics, such frameworks demonstrate that privacy and security are not fundamentally opposed objectives. Legitimate users can retain the privacy protections that encryption provides, while network defenders retain the visibility they need to detect and respond to threats.

As encrypted traffic volumes continue to grow and attack techniques continue to evolve, the development of lightweight, interpretable, and privacy-preserving detection methods will become increasingly central to the practice of network security. This work contributes a reproducible, well-validated, and practically grounded foundation upon which future research in this direction can confidently build.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," IEEE Access, vol. 6, pp. 33789–33795, 2018. https://doi.org/10.1109/ACCESS.2018.2841987

[2] R. Alshammari and A. N. Zincir-Heywood, "Investigating two different approaches for encrypted traffic classification," in Proc. 6th Annual Conference on Privacy, Security and Trust (PST), 2008, pp. 156–166.

[3] O. Barut, M. Grohotolski, C. DiLeo, Y. Luo, P. Li, and T. Zhang, "Machine learning based malware detection on encrypted traffic: A comprehensive performance study," in Proc. 7th International Conference on Networking, Systems and Security (NSysS), Dhaka, Bangladesh, 2020. https://doi.org/10.1145/3428363.3428365

[4] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," Journal of Computer Virology and Hacking Techniques, 2017. https://doi.org/10.1007/s11416-017-0306-6

[5] Canadian Institute for Cybersecurity, "CIC-IDS 2018 Dataset," University of New Brunswick, Fredericton, NB, Canada, 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[6] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in Proc. 3rd International Conference on Information Systems Security and Privacy (ICISSP), Porto, Portugal, 2017, pp. 253–262. https://doi.org/10.5220/0006105602530262

[7] S. Thaseen and A. K. Cherukuri, "Intrusion detection model using Chi-square feature selection and modified Naïve Bayes classifier," Springer Lecture Notes in Electrical Engineering, vol. 49, pp. 81–91, 2016.

[8] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 38, no. 5, pp. 649–659, 2008.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP), Madeira, Portugal, 2018, pp. 108–116. https://doi.org/10.5220/0006639801080116

[10] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, and J. Xu, "Malicious network traffic detection based on deep neural networks and association analysis," Sensors, vol. 20, no. 5, p. 1452, 2020. https://doi.org/10.3390/s20051452

[11] Google, "Encrypted traffic across Google," Google Transparency Report, 2020. [Online]. Available: https://transparencyreport.google.com/https/overview

[12] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 2009, pp. 1–6.

[13] D. M. Farid, H. Nouria, and M. Z. Rahman, "Combining Naive Bayes and decision tree for adaptive intrusion detection," International Journal of Network Security and Its Applications, vol. 2, no. 2, 2010. https://doi.org/10.5121/ijnsa.2010.2202

[14] European Parliament and Council of the European Union, "General Data Protection Regulation (GDPR)," Official Journal of the European Union, 2016. [Online]. Available: https://gdpr-info.eu

[15] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html