VNAT (VPN/NONVPN NETWORK APPLICATION TRAFFIC DATASET)

Ibrahim Alhusaini, Nasser Alhajri, Noor Hadari, Aseel Alridini, Rayyan Alnahwi

College of Computer Science and Information Technology, University of Dammam,
Kingdom of Saudi Arabia

2210002314@iau.edu.sa, 2210006797@iau.edu.sa, 2210002310@iau.edu.sa,
2210001894@iau.edu.sa, 2210002103@iau.edu.sa,

Abstract

The necessity to correctly detect VPN and Non-VPN traffic has expanded in importance because of escalating privacy worries and growing encryption usage during this period. The research examines how the VNAT (VPN/NonVPN Network Application Traffic) dataset can be analyzed through machine learning techniques for VPN and Non-VPN flow distinction when studying encrypted and unencrypted traffic patterns. The development objective targets an anticipatory model able to detect VPN usage accurately in situations where encrypted information exists or when applications evolve. A complete processing pipeline was established to prepare the data by cleaning it followed by duplicate removal after which garbage value treatment occurred alongside outlier clipping and SMOTE-based class balance implementation. Feature standardization together with normalization contributed to better model learning performance in algorithms. The research tested 11 supervised machine learning algorithms which evaluated datasets through phase 1 utilizing raw features then progressing to phase 2 with standardized features and finally phase 3 with both standardization and balanced normalization. Random Forest together with XGBoost and CatBoost delivered the most competent results for all measurements after conducting SMOTE and normalization. Among the best-performing models the F1-scores and accuracy exceeded 95% while showing strong detection capabilities toward the minority class (VPN traffic). The research highlights the necessity of both optimal data preparation techniques alongside suitable model selection strategies since this combination enables better intrusion detection and real-time monitoring equipment in secure networks.

Keywords: VPN detection, Non-VPN traffic, Network traffic classification, Encrypted traffic, Machine learning, Feature engineering, SMOTE, Supervised learning.

1. Introduction

Current internet usage increases together with data privacy concerns have driven widespread VPN selection in addition to encryption techniques for personal data protection. Standard VPN implementations deliver stronger security measures to their users yet such technologies are now being wrongly used by criminals to steal data and launch cyber attacks while they also help users evade security systems. The fields of network forensics together with IDS and cybersecurity monitoring require identifying and categorizing VPN traffic from Non-VPN traffic as a fundamental operational task.[25]

Traditional traffic classification approaches depended on port-based or payload inspection methods that became ineffective because encryption tunnels and alternative ports have become widespread. People in the research community have chosen to use machine learning (ML) approaches for identifying VPN and Non-VPN traffic by employing flow-level data analysis methods instead of working with packet contents. The majority of these investigative attempts relied on dated datasets while omitting real-world class distribution imbalances so their models became ungeneralizable due to their inherent bias.[25][26]

The proposed study implements a complete ML pipeline to analyze VPN and Non-VPN traffic flow records documented in the VNAT dataset that originates from various network applications and VPN protocols. Our research uses a three-step experimental design that includes benchmarking raw datasets and implementing standardization techniques with SMOTE-based sampling and MinMax normalization for creating well-balanced optimized datasets. A set of strict data preprocessing steps was applied to the dataset starting with duplicate removal before moving to outlier clipping and then treating garbage values and ending with feature scaling.[27]

The implementation of Support Vector Machine (SVM) occurred as one of eleven supervised ML classifiers with Random Forest (RF) and Decision Tree (DT) and K-Nearest Neighbors (KNN) and Artificial Neural Networks (ANN) and Logistic Regression (LR) and Naive Bayes (NB). Along with XGBoost and LightGBM and CatBoost and AdaBoost the eleven classifiers included boosting algorithms. The evaluation tracked three performance criteria including accuracy and precision and recall and F1-score metrics while placing emphasis on VPN traffic which was the minority event. The proposed approach succeeded in obtaining recall rates and F1-scores exceeding 95% after performing complete preprocessing and balancing techniques on the models.[26][28]

The study proceeds through its organization using this specific sequence. The section about reviewing existing literature appears in Section 2. Section 3 details both the proposed machine learning techniques while explaining their classifier structures. Section 4 includes both the dataset and experimental setup information for the empirical part of this research. The study concludes with two sections that display results and their discussions (Section 5) and the summary of final remarks and proposed recommendations (Section 6).

2. Review of Related Literatures

2.1 Early ML Approaches and Statistical Learning

The research conducted by Saber et al. (2018) established one of the earliest studies regarding encrypted traffic classification through hybrid machine learning models. A pipeline which integrated PCA with sampling techniques provided solutions for managing class imbalance together with high-dimensional data problems. Researchers applied their PCA-SVM methodology on ISCX VPN-nonVPN data to decrease the 23 initial features into 15 principal components. The testing of flow timeout variations yielded maximum performance at 15-second timing which produced 96.6% accuracy together with 0.97 precision and recall and exhibited minimal measurement deviation. The project developed methods to manage the relationship between feature dimensions and real-time system operations.

Shapira and Shavitt (2019) developed FlowPic which applied convolutional neural networks to grayscale images that resulted from traffic flow transformations for classification purposes. FlowPic develops a 2D image representation system based on packet sizes and inter-arrival times to allow pattern recognition through its LeNet-5 style CNN structure. When used on both ISCX VPN-nonVPN and Tor datasets the image-based method detected VPN traffic with an accuracy rate reaching 98.4%. The model displayed durability when used with different application networks and encryption protection systems. The research established traffic representation through images as an effective tool which surpasses the use of statistical features for detection purposes.

The researchers from Vu et al. (2019) presented an LSTM-based model which specialized in encrypted traffic time-series analysis. The research employed 55 time-based network flow features including packet dimension and interarrival time delays that they provided to a deep LSTM neural network model. The model processed 14,240 ISCX dataset samples to attain a 98% F1-score thus demonstrating better performance than MLP and SVM. This research demonstrated how packet series function well as time-series data and confirmed recurrent neural networks can effectively detect VPNs.

2.2 Emergence of Deep Learning and Parallel Networks

Traffic classification expanded into satellite communication areas according to Pacheco et al. (2020) who approached this challenging domain because of latency and bandwidth restrictions. The authors developed a hierarchical system which included handpicked features alongside deep learning capabilities through the use of 1D CNN architecture for the classification of regular and tunnelled internet protocols. The model received evaluation using both ISCX VPN-nonVPN and synthetic satellite datasets from OpenSAND. The CNN model achieved 99.18% accuracy which produced superior results than KNN and Decision Trees traditional classifiers. The study established that specialized communication environments require classification systems to be properly adjusted.

Bu et al. (2020) contributed NINparallel as a parallel deep learning model which implements Networ8k-in-Networ8k (NIN) layers. The system processed packet headers through one branch and packet payloads through another then merged them in later stages to extract thorough information from diverse input formats. NINparallel obtained 98.5% F1-score when testing it on 120,000 samples within a balanced subset of ISCX data. The authors highlighted how parallel feature abstraction helps encrypted traffic classification achieve better generalization results.

The SDN real-time classification was analyzed by Chang et al. (2020) in their study. The research group tested CNN together with MLP and SAE models for offline training using ISCX data to execute online inference through an SDN controller. The CNN model achieved offline accuracy of 94.86% yet its performance decreased to 87% when operated online due to delays introduced by SDN along with packet drops. The research examined the deployment obstacles that DL models face in fluid dynamic and real-time operational frameworks.

The authors Iliyasu and Deng (2020) developed a semi-supervised learning system through Deep Convolutional Generative Adversarial Networks (DCGANs). The researchers developed a model that cuts down the need for labeled data through creation of synthetic traffic samples with pseudo-labels. Using DCGAN generated results yielded an accuracy level of 89% on the QUIC dataset and 78% on ISCX when using only 10% of labeled training data. The research proved that GANs function well in conditions with limited network traffic classification resources.

2.3 Hybrid and Cascaded Models

In early 2021 Parchekani et al developed a two-step cascade architecture which combined MLP and LSTM components. This system divides the traffic into VPN and Non-VPN types before performing application classification on the Non-VPN flows. Using 118,848 ISCX and ISP traffic samples the model obtained 94% accuracy during assessment. Two-stage classification methods present successful results in organizing hierarchical flow labeling schemes according to this research study.

Sarhangian used a CNN-GRU deep learning combination to process both ISCX and private data sets throughout that same year. When processing 78 statistical elements extracted from flow-level data the system succeeded in binary classification with 99.23% accuracy while recording 68.57% accuracy in multi-class labelling. The research findings proved that hybrid deep networks deliver consistent performance when processing encrypted as well as heterogeneous network traffic.

The authors of Huoh et al. (2021) presented a Graph Neural Network (GNN) model where flows were represented as graphs containing temporal and packet attribute information. The GNN model performed 98.7% F1-score accurate VoIP detection while processing both 1,500-byte raw inputs with flow metadata information. In this application GNN maintained sequential packet ordering properties as well as flow structural dependencies.

At the same period Uğurlu et al. developed a minimal XGBoost-based system which adopted 15 time-oriented features sourced from the initial ISCX feature collection. Their model reached 94.53% accuracy after performing balancing along with tuning procedures. Time-optimized traffic classification becomes possible according to the results of their research.

Trang and Nguyen utilized ANN KNN and Random Forest as traditional ML algorithms for analyzing ISCX VPN-nonVPN traffic during 2021. The Random Forest technique achieved the best performance outcomes by reaching 94% accuracy during Scenario A1 (between VPN and Non-VPN traffic). ANN demonstrated session duration independence which makes it suitable for changing classroom conditions. The researchers developed a model to lower dependency on labeled data by producing synthetic pseudo-labeled traffic samples. The performance evaluation of DCGAN used both QUIC and ISCX datasets where it reached 89% accuracy while handling 10% labeled data on QUIC datasets and 78% accuracy on ISCX datasets. GANs proved their capability to perform traffic classification tasks under limited resource conditions.

2.4 Real-Time and Ensemble Techniques

Almomani launched the stacking ensemble model for 2022 which unified Random Forest and SVM and Artificial Neural Networks through a Logistic Regression meta-classifier. The system received training using 43,191 samples from ISCX VPN-nonVPN while utilizing 61 selected features in post-processing before evaluation. The developed model exceeded all individual models by reaching 98.87% accuracy in its performance. The research demonstrated that performance robustness can be achieved through combining different models while employing fusion strategies particularly in encrypted multi-class traffic systems.

Al-Fayoumi et al. (2022) performed a feature evaluation with Random Forest classifiers for VPN and Non-VPN traffic classification. Using the ISCX dataset they explored 15,545 samples with time-based flow features that included flow duration and inter-arrival times and packet rates. The model achieved 95.02% accuracy using all 24 features but the selected features through Pearson correlation with genetic algorithms maintained competitive results. The research demonstrated how time-based traffic behavior remains essential for making accurate VPN detection decisions.

In their work Izadi et al. (2022) presented a Bayesian decision-level fusion framework to function as an ensemble model. The authors used a hybrid model which integrated output results from CNN and DBN and MLP to exploit specific advantages of each component. Accurate classification reached 97% effectiveness with the fusion model through training it on the ISCX VPN-nonVPN dataset containing 890,000 flows. The research concluded that ensemble fusion stands as a strong method to enhance classifier resistance when facing class imbalance and noise conditions.

Goel et al. (2022) published research aimed at operational deployment of ML models within cloud-based infrastructure at the same moment. The detection system that they built operated within AWS infrastructure by analyzing statistics from Wireshark and NetMate flow data. The dataset contained 383,889 samples that originated from both live traffic and ISCX environments.

Random Forest proved to be the most effective model between those tested with a 95.43% accuracy level that makes it ideal for real-time VPN detection. The VPN blocking capability of the system created an important policy enforcement feature which operated in real time.

2.5 Optimization-Focused and Cost-Sensitive Models

Using XGBoost with Focal Loss approaches and Bayesian tuning Cao et al. (2022) developed a cost-sensitive solution to deal with feature redundancy alongside class imbalance problems. The model used mRMR for feature selection and received training through 37,028 samples from ISCX dataset. The adapted loss functions produced powerful results by enabling a detection model to achieve 97.43% accuracy when handling VPN traffic.

Ergönül and Demir (2022) introduced LSTM-FS as a real-time classifier which analyzes flow-level packet headers with LSTM networks during online detection. Their VPN traffic monitoring system received evaluation using a massive ISCX dataset that contained more than 25 million packets. The model reached 99.85% accurate online detection results at processing speeds which exceeded milliseconds and operated with high throughput performance. The implementation of LSTM-FS made it one of the most usable end-to-end systems for real-time encrypted traffic classification in operational networks.

The XGBoost optimization received continued development from Cicioğlu et al. (2023) who managed a XGBoost implementation using 889,808 samples derived from the ISCX dataset. Multiple feature selection algorithms with Chi-Square and ANOVA F-tests were used to decrease the original 93 features down to 42 which enhanced both operational speed and interpretability for the model. Their F1-score metrics reached 99.38% after tuning with FS+CV techniques which placed their model among the best in the field. A structured optimization pipeline stands essential for developing powerful classifiers with good scalability according to their research findings.

2.6 Image-Based and Novel Representation Techniques

Sun et al. (2023) developed a CNN-based model which employed “Packet Block Images” derived from combining packets originating from similar paths into 2D visual elements. The researchers tested their OpenVPN dataset model with 5,744 samples and reached 97.2% accuracy. Visual representations pioneered by FlowPic received further development through this methodology which produced excellent classification outcomes when analyzing highly encrypted data sets.

The research of Jorgensen et al. (2023) creates a machine learning system for encrypted traffic classification that puts uncertainty measurement and model generalization at its core. The researchers implemented Prototypical Networks on the VNAT dataset that combined VPN and non-VPN network flows from five application categories to extract statistical and wavelet-based features. Tests of the model revealed excellent performance markers including 0.98 micro F1-score alongside remarkable out-of-distribution settings along with 0.97 F1 maintenance after padding packets. The system displays excellent features for transfer learning together with low-data needs which makes it suitable for real-world situations and environments with evolving networks.

2.7 Hybrid and End-to-End Systems

Miller et al. presented a practical application for a lightweight MLP model which detects VPN traffic in real time during 2023. The researchers processed 3,952 custom flows while using 13 selected features derived from NetMate. Test results demonstrated the MLP classifier produced 98.42% accuracy while the researchers distinguished their work by deploying their solution on AWS-hosted systems.

Koumara et al. developed NetTiSA as a light-weight time-series-based flow feature set for high-speed network traffic classification in year 2023. The system reaches 100 Gbps deployment speeds through its application of 20 features which track packet timing together with flow symmetry measurements. The NetTiSA system reached binary classification accuracy between 99.96% and multiclass accuracy at 99.8% through tests on 15 public datasets across 25 tasks including VPN and botnet detection while using only 113 bytes for telemetry data. The ipfixprobe implementation of NetTiSA achieved 200K flows per second processing speed which surpassed the capabilities of CICFlowMeter in terms of speed as well as functionality. The network traffic classification approach SFTS was presented by Tomáš Čejka in 2023 as a time-series-based methodology which collects 69 features from whole streaming data instead of relying on minimal packet samples. These features divide their domains among five aspects and conducted their tests on 15 publicly accessible datasets for 23 different classification tasks related to VPN detection and botnet and IoT attack identification. The XGBoost model trained by HyperOpt reached 99.98% accuracy during VPN - VNAT detection. Real-time implementation of SFTS remains effective and accurate even when using just 10 features because it achieves accuracy only slightly below its initial score by 0.1%.

Huang et al. carried out a large-scale research on Residential IP Proxy (RESIP) traffic detection in 2024 as this type of proxy has seen rising usage for both hiding identities and harmful purposes. A team obtained 3.3 terabytes of flow data amounting to 116 million records from operational RESIP services to construct the classifiers for identifying relayed and tunnelled traffic. The study evaluated models through a Random Forest (RF-RF) supported by 186 handcrafted features and the combination between Random Forest (RF) and BERT Transformer. The RF-RF model reached 96.20% precision and 88.80% recall when using the first four upstream packets for detection at more than 298 flows per second. The detection system successfully applies to external datasets VNAT and ISCX2016 so it demonstrates suitability for deployment at Internet Service Provider gateways.

The research by Mohamed and Kurnaz (2024) presented a joint network of Artificial Neural Networks and XGBoost ensemble which underwent training utilizing the ISCX VPN-nonVPN dataset. Time-aware flow features enabled the hybrid system to reach 98.79% accuracy levels.

The research focused on establishing how deep non-linear models should be integrated with interpretable gradient boosting when analyzing encrypted traffic flows.

The research work of Elmaghraby et al. (2024) demonstrated ensemble modeling by processing sessions with multiple classifiers such as RF together with SVM and KNN as part of an integrated system. The system used 33,286 TCP sessions for training which resulted in 99.6% accuracy as the highest recorded achievement. Such a detection system applied ensemble voting methods on session-based traffic groups which improved VPN recognition abilities specifically targeting VoIP and streaming protocols.

Researchers at Babaria et al. presented FastFlow in 2025 as a fast and accurate traffic classifier based on LSTM architecture combined with reinforcement learning for early and precise traffic identification. FastFlow demonstrates high accuracy in traffic classification through its merger of granular packet measurement and gross slot input analysis which allows it to analyze 8 packets and 0.5 seconds of traffic to achieve over 91% accuracy. The model includes synthetic outlier training for unknown traffic detection and shows superior performance than previous models such as GGFast and Grad-BP with faster and more robust operations.

2.8 Gap Analysis

The quick evolution of encrypted traffic classification methods during recent years has propelled significant improvement of machine learning (ML) and deep learning (DL) technology applications in this domain. Multiple important deficiencies still exist which prevent the practical application and extended use of modern security solutions. The main problem stems from creating models that maintain stability and universal applicability across real-life operation environments. Various studies employing CNNs and LSTMs together with ensemble methods succeed in achieving high accuracy scores when working on ISCX VPN-nonVPN benchmark datasets. These models develop a specialization for particular traffic patterns in lab environments which produces poor results when dealing with network evolution and emerging security protocols combined with new applications. The lack of adaptable framework systems remains a challenge because DCGANs and Prototypical Networks do introduce low-data learning techniques however these solutions do not provide continuous adaptation via dynamic traffic changes without needing regular updates or large datasets.

The critical gap exists between achieving high accuracy with optimal computational performance and time-sensitive applications being possible at once. Deep neural models that incorporate hybrid CNN-GRU architecture together with Transformer systems require high processing power that prevents their deployment in quick networks or constrained systems including IoT gateways. New lightweight detection systems such as FastFlow, NetTiSA and XGBoost-based classifiers have entered the market to tackle efficiency problems yet they reduce their flexibility due to which they cannot process complicated multi-class situations and identify complex VPN or proxy tunneling procedures. Several XGBoost and Random Forest implementations use static feature selection methods and handcrafted features which creates difficulties in system adaptability when dealing with encrypted traffic obfuscation methods that involve packet padding as well as dynamic port usage and protocol mirroring.

Most research on VPN flow classification is dedicated to achieving high accuracy rates under optimal conditions yet fails to prioritize explainability and interpretability aspects alongside policy integration. When managing networks for cybersecurity purposes, stakeholders need both precise predictions about VPN and non-VPN classifications along with explanations of prediction reasons to use in compliance audits and response planning procedures. Minimal research exists about the integration of explainable AI (XAI) into encrypted traffic classifiers resulting in trust and usability problems for operational implementation. The deployment of cloud-based systems received attention by Goel et al. (2022) yet no complete end-to-end system exists which connects detection models to automated network enforcement systems as well as threat intelligence feeds.

A notable absence exists regarding the identification of new security risks and unorthodox encryption methods which extend past typical VPN traffic structures. Forces such as Residential IP Proxies (RESIP), peer-to-peer VPNs and stealth tunneling techniques process a dynamic threat environment that makes static models ineffective. Recent commercial research into RESIP detection stands out but more comprehensive solutions should be developed to detect entire ranges of encrypted traffic misuse exclusively for zero-day tunneling approaches and adaptable evasion methods. Research should create classification systems which maintain high performance while providing effective insights for network administrators to operate in advanced encrypted digital platforms.

Reference	**Dataset**	**# Samples**	**# Features**	**Technique (Best)**	**Result (Accuracy)**
[33]	UNIBS, VNAT	22.9million	Packet-level: direction, size, IAT	LSTM	98.03%
[32]	RESIP relayed/tunnel flows + VNAT + ISCX2016	12,000 labeled	186 handcrafted	RF (with features)	96.20%
[7]	Custom	33,286	20	RF (Ensemble)	99.60%
[17]	ISCX	Not stated	12	ANN + XGB	98.79%
[24]	Custom	3,952	13	MLP	98.42%
[21]	OpenVPN + ISCX	ISCX: ~1.9k, OpenVPN: ~5.7k+	Image blocks	CNN	97.20%

[31]	VNAT, ISCX, CTU-13	Up to 10M+ flows	20	XGBoost	99.96%
[30]	VNAT, ISCX, CTU-13)	Not stated	69 full (reduced to 10)	XGBoost	99.98%
[26]	ISCX	889,808	42	XGBoost	99.38%
[29]	VNAT, PCAP	33,711	129	Prototypical Networks + OOD score	98%
[9]	ISCX	25M+	17	LSTM	99.85%
[10]	ISCX	43,191	61	RF+SVM+ANN+LR	98.87%
[28]	ISCX + Live	383,889	18	Random Forest	95.43%
[18]	ISCX	37,028	23	FL-XGB	97.43%
[23]	ISCX	890,000	23	CNN+DBN+MLP	97.00%
[27]	ISCX	15,545	24	Random Forest	95.02%
[11]	ISCX + ISP	118,848	784	MLP + LSTM	94.00%
[14]	ISCX	27,695	Raw+Meta	GNN	98.70%
[16]	ISCX	Not stated	15	XGBoost	94.53%
[12]	ISCX	Not stated	Not stated	Random Forest	94.00%
[15]	Private + ISCX	92,889	78	CNN-GRU	99.23%
[19]	ISCX	22,181	305	CNN	94.86%
[25]	ISCX + QUIC	115K+	3	DCGAN	89.00%
[22]	ISCX	120,000	Raw bytes	NINparallel	98.50%
[6]	ISCX + Emulated	Multiple	38–336	1D CNN	99.18%
[20]	ISCX	14,240	55	LSTM	98.00%
[13]	ISCX + Tor	Not stated	1500x1500	FlowPic CNN	98.40%
[8]	ISCX	Not stated	15	PCA-SVM	96.60%

*Table 1: **Summary of Literature Review***

3.0 Material & Methods:

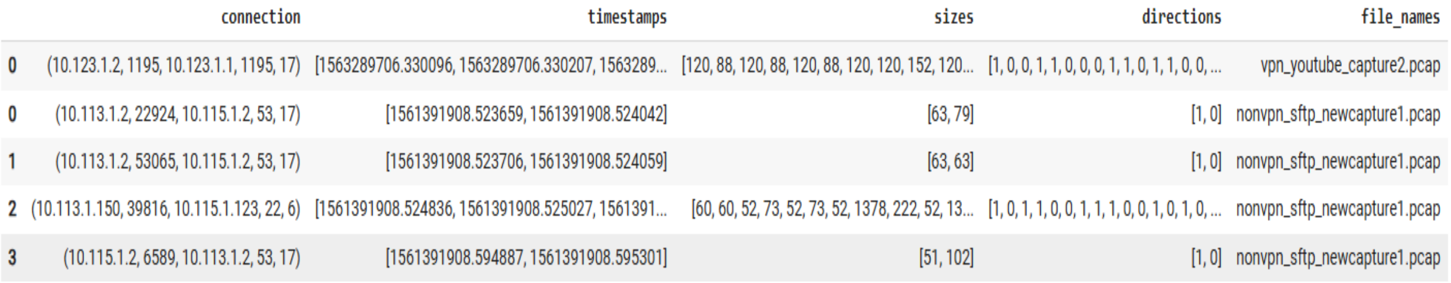
The following part explains the tools and datasets as well as the techniques that were employed for VPN and Non-VPN encrypted network traffic classification. The inaugural section explains the data origination alongside its organization approach before proceeding to demonstrate data preparation and feature transformation steps. The document provides details about machine learning algorithm testing and performance evaluation metrics as well as the justification for selecting specific models.

3.1. Description of the Dataset

The research draws its data from the VPN/NonVPN Application Traffic (VNAT) dataset which MIT Lincoln Laboratory established. The data collection spanned 272 hours during which 33,711 traffic flows from real applications were monitored under VPN and non-VPN settings. The dataset includes five major categories which consist of streaming, VoIP, chat, file transfer along with command & control (C2). Each traffic flow contains time stamps along with network packet features and tunnel encryption designators along with direction information. The research required conversion of the HDF5 format to flat CSV and a binary label system was established (VPN labeled 0 while Non-VPN equaled 1). The flow-duration is the first important feature in the dataset followed by packet-count and total-size and incoming-count and outgoing-count and direction-ratio. The distribution of records showed extreme class imbalance since only 1.1% of all flows were labeled as VPN thus requiring data resampling approaches.

3.1.1 Statistical Analysis of the Dataset

3.1.1.1 Initial Sample Records from VNAT Dataset Before Preprocessing



	connection	timestamps	sizes	directions	file_names
0	(10.123.1.2, 1195, 10.123.1.1, 1195, 17)	[1563289706.330096, 1563289706.330207, 1563289...	[120, 88, 120, 88, 120, 88, 120, 120, 152, 120...	[1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, ...	vpn_youtube_capture2.pcap
0	(10.113.1.2, 22924, 10.115.1.2, 53, 17)	[1561391908.523659, 1561391908.524042]	[63, 79]	[1, 0]	nonvpn_sftp_newcapture1.pcap
1	(10.113.1.2, 53065, 10.115.1.2, 53, 17)	[1561391908.523706, 1561391908.524059]	[63, 63]	[1, 0]	nonvpn_sftp_newcapture1.pcap
2	(10.113.1.150, 39816, 10.115.1.123, 22, 6)	[1561391908.524836, 1561391908.525027, 1561391...	[60, 60, 52, 73, 52, 73, 52, 1378, 222, 52, 13...	[1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, ...	nonvpn_sftp_newcapture1.pcap
3	(10.115.1.2, 6589, 10.113.1.2, 53, 17)	[1561391908.594887, 1561391908.595301]	[51, 102]	[1, 0]	nonvpn_sftp_newcapture1.pcap

Figure 1: first five rows of original dataset

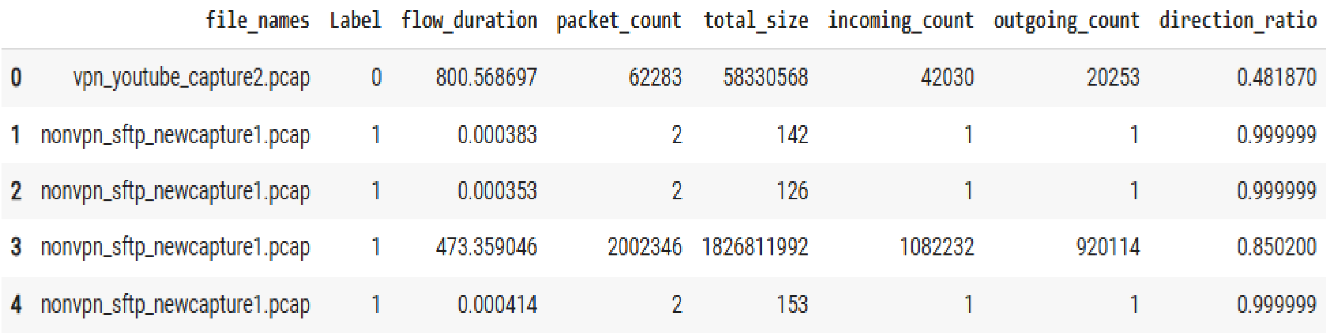
The initial display demonstrates the first five rows of original .h5 dataset prior to processing or flattening operations. A network flow occupies each row in this dataset while presenting information through connections between source/destination IPs and ports with protocol and timestamps for packet arrival together with sizes in bytes and directions as binary flags and file_names indicating VPN or Non-VPN source status.

The initial unprocessed format establishes the foundation for upcoming attribute retrieval together with category determination operations.

After initial examination the raw VNAT dataset reveals multiple attributes inside its nested structure that prevents machine learning application because the packet timestamps and their sizes and directions are not directly usable for these tasks. A transformation through flattening processed this unorganized data to make it suitable for analysis. The file_names column provided the basis for creating binary labels by assigning value 1 to "nonvpn" flows and 0 to all other cases. After that the system extracted meaningful numerical data points from headings contained inside list structures. The flow duration calculation depended on subtracting maximum timestamp value from minimum timestamp value across each flow. The packet count derived from a count of all sizes list values while the total size required an addition of these values together. Directional behavior information was derived from flow data through a process which generated the ratio between outgoing (1) and incoming (0) packets in each flow.

The final set retained flattened numerical features together with the binary label and the file name reference after discarding the transformed connection, timestamps, sizes and directions columns. The finalized dataset received the name VNAT_binary_flatten.csv as it underwent flattening for further preprocessing and machine learning processing.

3.1.1.2 first five rows for original dataset

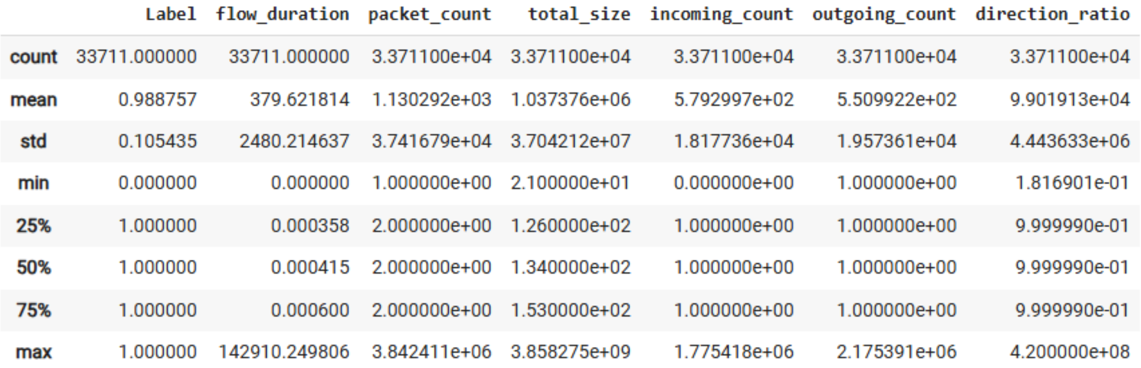


	file_names	Label	flow_duration	packet_count	total_size	incoming_count	outgoing_count	direction_ratio
0	vpn_youtube_capture2.pcap	0	800.568697	62283	58330568	42030	20253	0.481870
1	nonvpn_sftp_newcapture1.pcap	1	0.000383	2	142	1	1	0.999999
2	nonvpn_sftp_newcapture1.pcap	1	0.000353	2	126	1	1	0.999999
3	nonvpn_sftp_newcapture1.pcap	1	473.359046	2002346	1826811992	1082232	920114	0.850200
4	nonvpn_sftp_newcapture1.pcap	1	0.000414	2	153	1	1	0.999999

Figure 2: first five rows of flatten dataset

The first few rows of processor-ready data are shown in the figure. The extracted numerical features found in each row of the network flow data include flow_duration, packet_count, and total_size with their respective incoming and outgoing packet counts. The direction_ratio helps identify the degree of traffic movement between the networks. The Label column contains two target classes where VPN traffic is tagged with value 0 and Non-VPN traffic is assigned value 1. The structured data design improves machine learning model training and evaluation processes for VPN/Non-VPN classification tasks.

3.1.1.3 data descriptive statistics



	Label	flow_duration	packet_count	total_size	incoming_count	outgoing_count	direction_ratio
count	33711.000000	33711.000000	3.371100e+04	3.371100e+04	3.371100e+04	3.371100e+04	3.371100e+04
mean	0.988757	379.621814	1.130292e+03	1.037376e+06	5.792997e+02	5.509922e+02	9.901913e+04
std	0.105435	2480.214637	3.741679e+04	3.704212e+07	1.817736e+04	1.957361e+04	4.443633e+06
min	0.000000	0.000000	1.000000e+00	2.100000e+01	0.000000e+00	1.000000e+00	1.816901e-01
25%	1.000000	0.000358	2.000000e+00	1.260000e+02	1.000000e+00	1.000000e+00	9.999990e-01
50%	1.000000	0.000415	2.000000e+00	1.340000e+02	1.000000e+00	1.000000e+00	9.999990e-01
75%	1.000000	0.000600	2.000000e+00	1.530000e+02	1.000000e+00	1.000000e+00	9.999990e-01
max	1.000000	142910.249806	3.842411e+06	3.858275e+09	1.775418e+06	2.175391e+06	4.200000e+08

Figure 3 table summary of statistics dataset

The statistical analysis done to the flattened VNAT dataset reveals important details about feature distributions together with feature variability patterns. The flow_duration feature contains values extending from 0 seconds to 142,910 seconds (~39 hours) with a mean time of 380 seconds but demonstrates a very high standard deviation of 2480 seconds caused by extreme outliers and distribution skewness. The maximum values for packet_count and total_size reach 3.8 billion in total size and 3.8 million packets even though the median values remain minimal. The significant gap between mean and median values together with high standard deviations show that traffic flows have substantial skewness and contain possible abnormal traffic patterns. The majority of traffic flows have minimal packet counts combined with minimal incoming/outgoing packets as well as small packet sizes because most samples contain a small number of packets. The direction_ratio exhibits abnormally wide variation from 0.18 to more than 420 million due to division operations involving low or zero packets that cause instability in the ratio metric. The inconsistent data patterns require preprocessing methods such as outlier correction with normalization techniques and maybe feature normalization to safeguard learning accuracy when delivering data into machine learning models.

3.1.2 Data Exploratory Analysis (EDA)

3.1.2.1 Bar chart the distribution of the target label before pre-processing

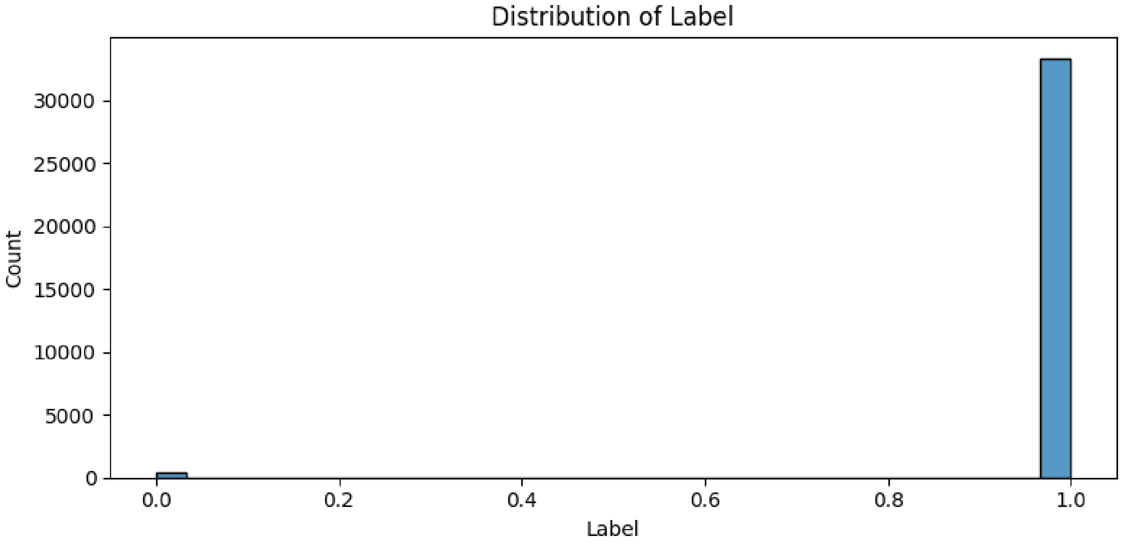


Figure 4: distribution histogram of label class

The provided chart demonstrates significant data imbalance because Non-VPN samples outnumber VPN samples by a wide margin. Machine learning models face challenges when dealing with class distribution imbalance because these models tend to predict the more abundant category which is Non-VPN. The data requires balancing through sampling techniques such as SMOTE or undersampling which will optimize classification results.

3.1.2.2 Box Plot Analysis of Numerical Features

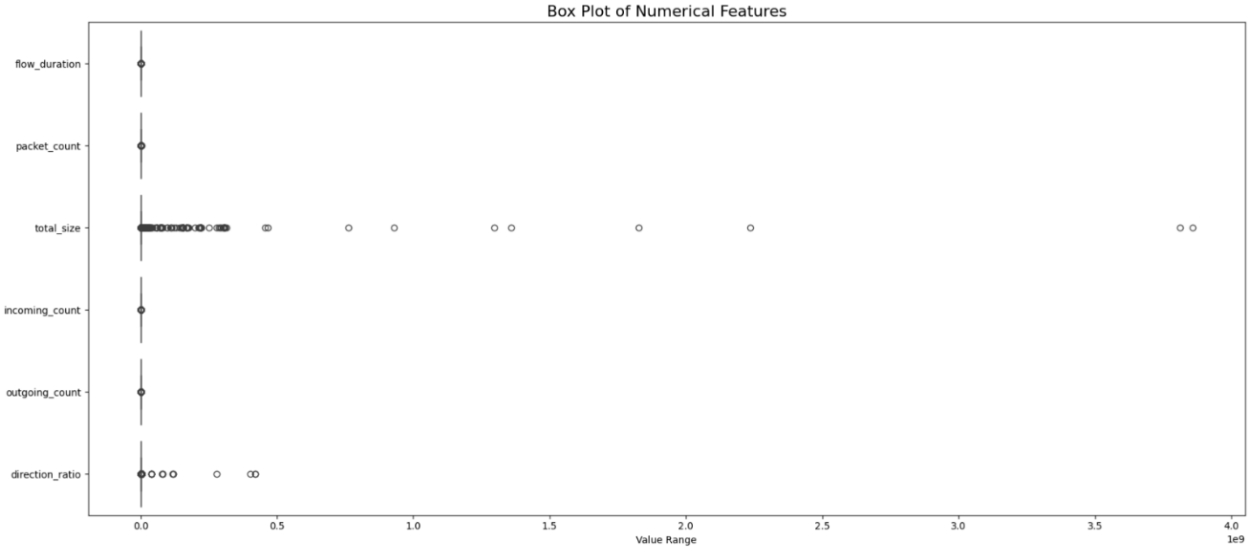


Figure 5: box plot to detect outliers

A grouped box plot was made to examine together side by side all numerical features in the VNAT dataset while presenting their distribution and variability data. A clear distribution pattern shows extensive outliers affect most variables in the dataset with flow_duration, packet_count and total_size containing long tails and numerous extreme data points. Network traffic measurements demonstrate this regular characteristic because numerous flows stay brief while bigger or longer sessions appear at higher levels in the distribution. The wide range of values in the direction_ratio feature exists because of its derivative calculation method which produces errant results. The plot demonstrates non-normal data distribution and feature distribution imbalance which confirms that data normalization and outlier treatment are needed before training machine learning models.

3.1.2.3 Heatmap visualizes the correlation matrix between the numerical features in the flattened VNAT dataset

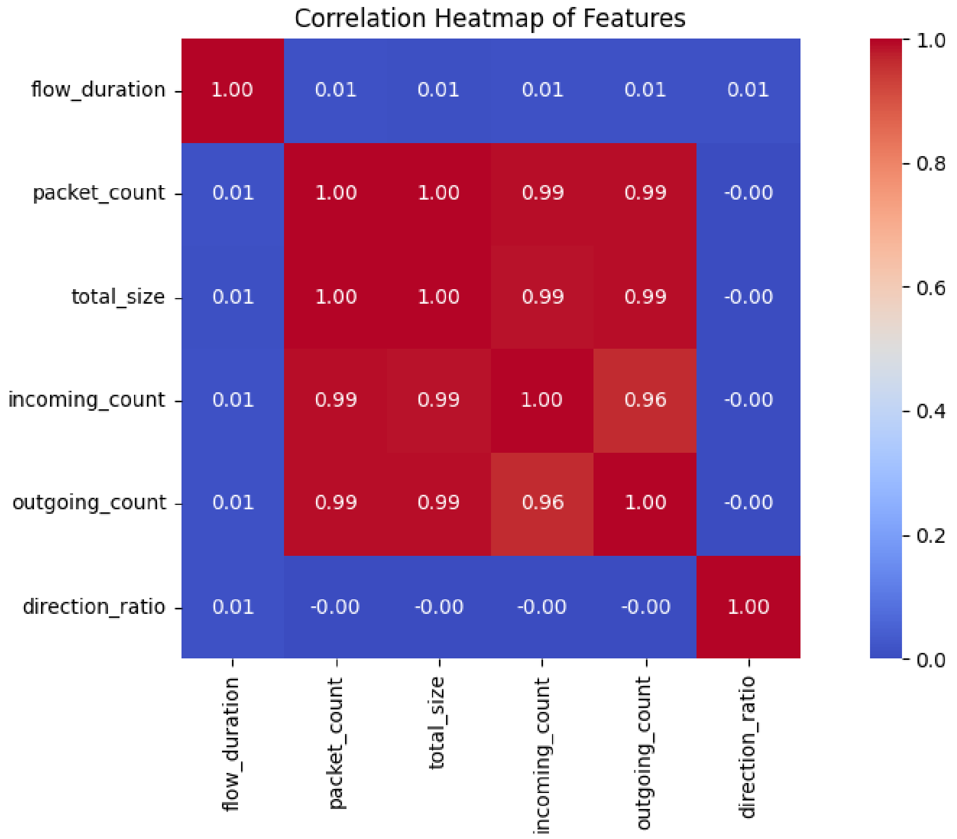


Figure 6: heatmap of correlation

The simultaneous relationship between size and count variables leads to degraded model performance while causing increased variance among linear models. Feature selection through techniques such as PCA should be implemented to eliminate this feature redundancy. The model's generalization capabilities will benefit from retaining direction_ratio and flow_duration variables even though they show no correlation.

3.2. Preprocessing

3.2.1 Label Generation and Flattening of Nested Features

The original .h5 dataset included nested information grouping information on IP connections with timestamps and packet sizes and directions into single structures. Our processing made the data structures flat to support classical machine learning model reading capabilities. The flow-duration and packet-number and total-data-size and incoming-packet-number alongside outgoing-packet-number and direction-relationship were added as new numerical measures. A binary variable was established for VPN traffic using 0 as VPN traffic labels and 1 as Non-VPN traffic indications based on file_name analysis. The system established conditions which enabled the use of standardized machine learning pipelines.

3.2.2 Dataset Standardization

After initial evaluations of unmodified features we standardized the dataset to match all quantitative values on the same measurement range. Standardized data offers improved output from most machine learning tools including SVM, KNN and Logistic Regression through preventing size scale issues from affecting individual factors. Standard deviation division followed by mean subtraction performs the standardization of features.

3.2.3 Duplicates & Garbage Value Treatment

The data quality assurance process began by conducting a thorough examination that allowed us to find duplicate entries and eliminate possible incorrect values. Model training suffers from sizable bias creation and distorted learning outputs that diminish generalization effectiveness because of duplicate rows if these records remain unchanged. During preprocessing errors and statistical measures become skewed when garbage values such as ill-formed inputs or incompatible field structures or infinite values are detected. The analyzed dataset contained 18,653 duplicate records which were detected.

The repeated data points went through a safe removal process ensuring each data entry played an individual role in teaching the system. The step served to eliminate any concealed noise and data contamination even though it contained no significant garbage values (NaN, inf, or structurally invalid types) thus validating its importance. The clean-up process increased the data reliability level before we conducted meaningful statistical analysis and built stable learning algorithms.

3.2.4 Eliminating Constant Features

We checked all rows to identify features which held identical values throughout. Such features appear consistently across all values which makes them uninformative and results in extra dimensions in the dataset. The model runs faster and maintains accuracy levels despite the removal of these features.

3.2.5 Handling Missing Values

The analysis checked whether NaN values existed in the dataset because they could affect either model performance or training stability. The analysis evaluated the size of missing data points which received either appropriate value imputation techniques (such as median or mean) or row removal when the values were extensive and rare.

3.2.6 Handling Outliers with IQR Clipping

We used IQR clipping across every numerical feature as a method to fight extreme values. The IQR thresholds on each column became 1.5 times the IQR range with the result that outlier values were truncated. The model benefited from this technique that eliminated the impact of exceptionally large values present in flow_duration and total_size features on the training process.

3.2.7 SMOTE Class Balancing

A review of the preprocessed class distribution showed an enormous sample imbalance between VPN and Non-VPN categories. Among the 34,011 analyzed samples, only 379 samples made up 1.12% of VPN usages but the remaining 33,332 samples formed 98.88% of the Non-VPN category. The machine learning models would likely produce suboptimal detection of the minority (VPN) class because of this severe class distribution imbalance.

We solved this problem by implementing SMOTE (Synthetic Minority Over-sampling Technique) as a solution. The minority class in SMOTE receives synthetic copies of instances which originate from matching features between native examples. The SMOTE over-sampling process creates new samples from the minority class while preventing redundant instances that may result in overfitting. The classifiers gained the ability to accurately represent two classes after the application of SMOTE because it balanced the classes while improving model fairness and accuracy.

3.2.8 MinMax Normalization

The normalization to range [0, 1] through MinMax normalization followed the SMOTE process. Post-balancing the features maintained similar values which proved essential for KNN distance-based algorithms.

3.2.9. Feature Review and Correlation Analysis

We generated a correlation heatmap to understand the level of relationship among features in addition to their multicollinearity effect. The linear relationships between packet-count and total-size together with direction counts indicated repetition of information. Further selection of features or dimensional reduction methods should consider these findings in their progression.

3.3 Feature Engineering

Feature Engineering uses methods for selecting existing data features alongside transforming and building new one from data to enhance ML model performance. This approach studies domains and data structure together with problem definition in order to develop features which help models detect crucial data relationships. The process contains two basic methods which include selecting specific data features alongside creating new available data features.

3.3.1 Feature Extraction

The initial dataset contained intricate nested lists with three major components such as packet sizes, direction values and timestamp measurements. These unprocessed data elements required flattening in addition to transformation before becoming suitable for model input because the raw forms did not suit direct use. A series of numerical features were obtained from the nested fields. A set of relevant numerical characteristics served as the extracted features:

- flow_duration: duration of a flow (max timestamp – min timestamp)
- packet_count: the total number of packets in the flow
- Total size: total of all packet sizes in the flow
- incoming_count and outgoing_count: number of packets in each direction
- direction_ratio: proportion of packets outgoing to incoming packets

Through these extracted features the traffic data was converted into tabular format from its complex sequential structure to enable traditional machine learning application methods. The Label received a binary format during this stage to identify VPN traffic through 0 values and Non-VPN traffic through 1 values.

3.3.2 Feature Selection

Two main reasons prevented us from employing direct feature selection techniques including Recursive Feature Elimination (RFE) as well as correlation-based filtering and model-based importance ranking. The preprocessed dataset contained between 7–8 specifically engineered features therefore the training process could be managed efficiently and safely without overfitting. Each feature in this system was developed with the purpose to reveal vital VPN/Non-VPN traffic identification information. The model used all the developed features within its training process.

3.4 Post-Processing Data Visualization

3.4.1 Label Distribution After SMOTE Balancing

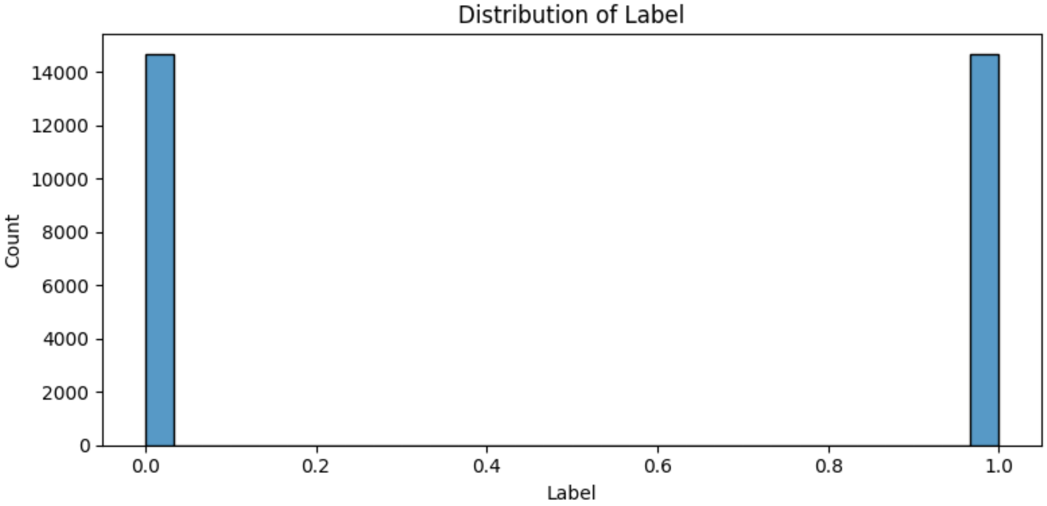


Figure 7: after balance

The SMOTE enhancement technique revealed a bar plot depicting equalized VPN and Non-VPN classes. The implementation of each class with 14,700 samples addresses the initial problem that very few samples belonged to the Non-VPN category. Balancing the target classes prevents model bias and enables effective learning of symmetric data patterns from both samples.

3.4.2 Box Plot of Label After SMOTE

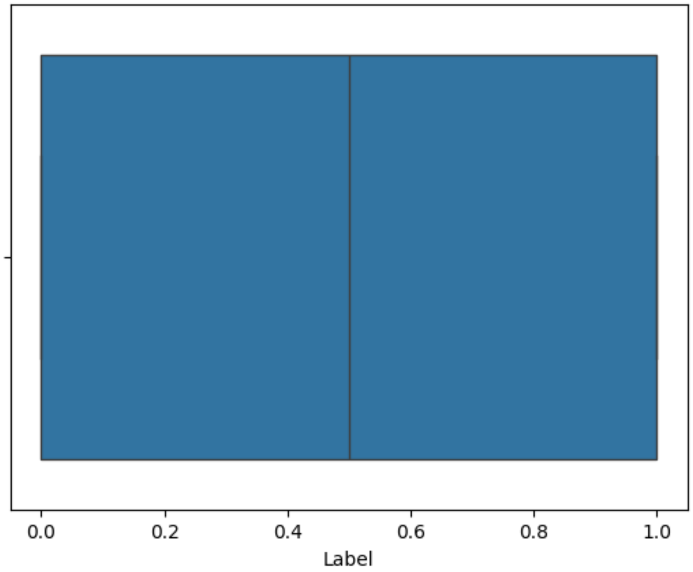


Figure 8: box plot after balance

The box plot demonstrates the two-class label distributions after executing SMOTE (Synthetic Minority Oversampling Technique). Testing reveals that both classes (0 and 1) are equally distributed with matching interquartile ranges and no outliers after SMOTE oversampling.

3.4.3 Box Plot of Numerical Features After Preprocessing

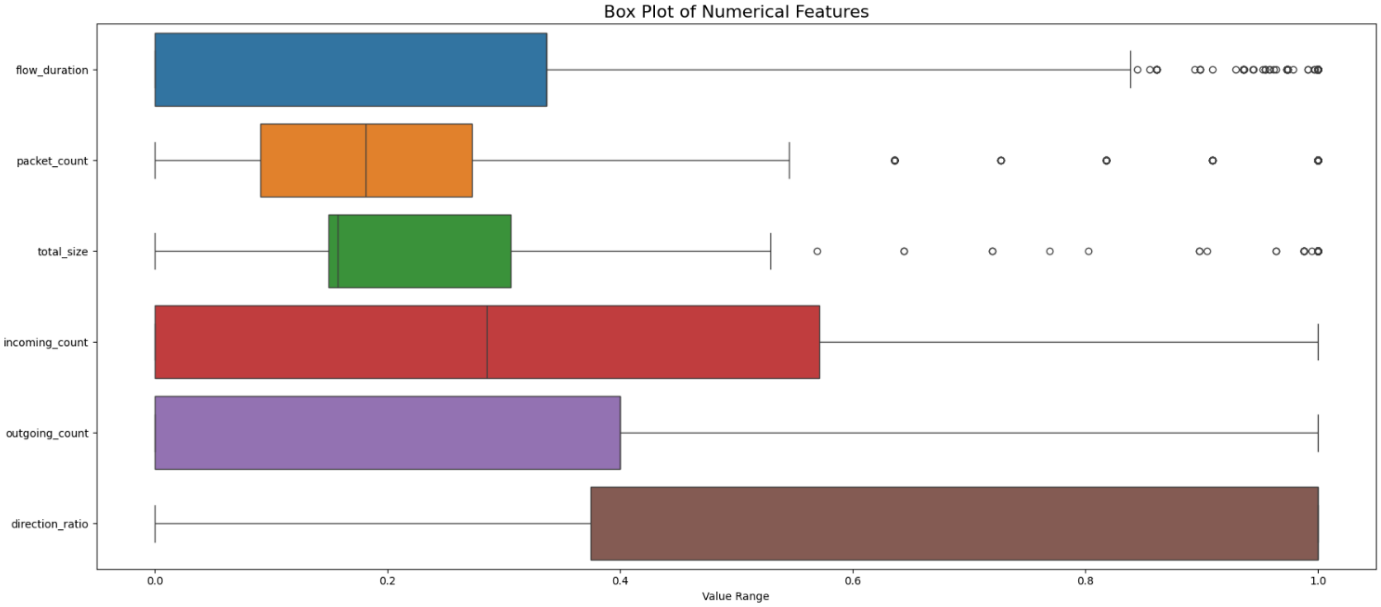
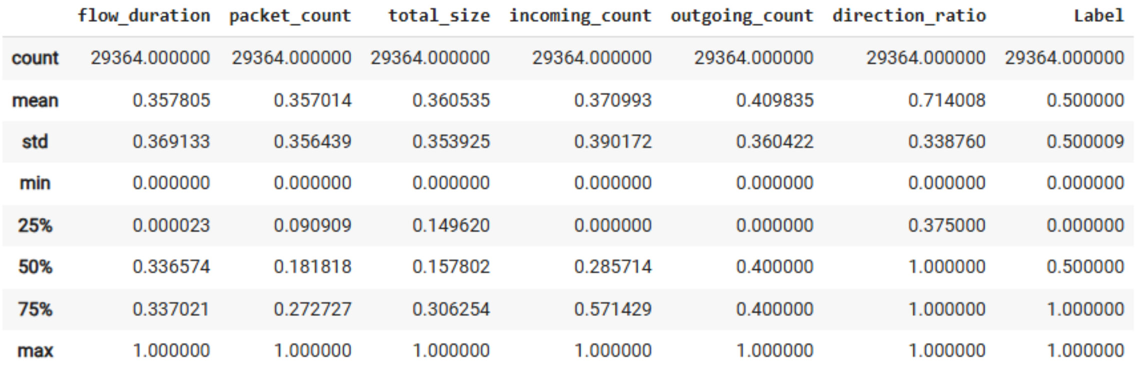


Figure 9: box plot for numerical features

A box plot shows the numerical distribution of data throughout all preprocessing procedures which consisted of outlier handling, normalization steps and SMOTE sample creation. Outlier effects on flow_duration, packet_count and total_size have been reduced to a minimum level. The dataset obtained its reliable machine learning model training capability because normalization normalized feature scales to achieve consistent metrics throughout all data points**.**

3.4.4 Descriptive Statistics After Normalization and Balancing



	flow_duration	packet_count	total_size	incoming_count	outgoing_count	direction_ratio	Label
count	29364.000000	29364.000000	29364.000000	29364.000000	29364.000000	29364.000000	29364.000000
mean	0.357805	0.357014	0.360535	0.370993	0.409835	0.714008	0.500000
std	0.369133	0.356439	0.353925	0.390172	0.360422	0.338760	0.500009
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000023	0.090909	0.149620	0.000000	0.000000	0.375000	0.000000
50%	0.336574	0.181818	0.157802	0.285714	0.400000	1.000000	0.500000
75%	0.337021	0.272727	0.306254	0.571429	0.400000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 10

The summary table displays the dataset with 29,364 instances that have normalized values and SMOTE-balanced distribution while maintaining equal VPN and Non-VPN classes. The input ranges for model training are consistent because all features have received a scale transformation from 0 to 1. The dataset displays uniform statistical metrics between features with balanced label distribution which ensures dependable machine learning operations.

3.4.5 Correlation Heatmap of Normalized Features After Preprocessing

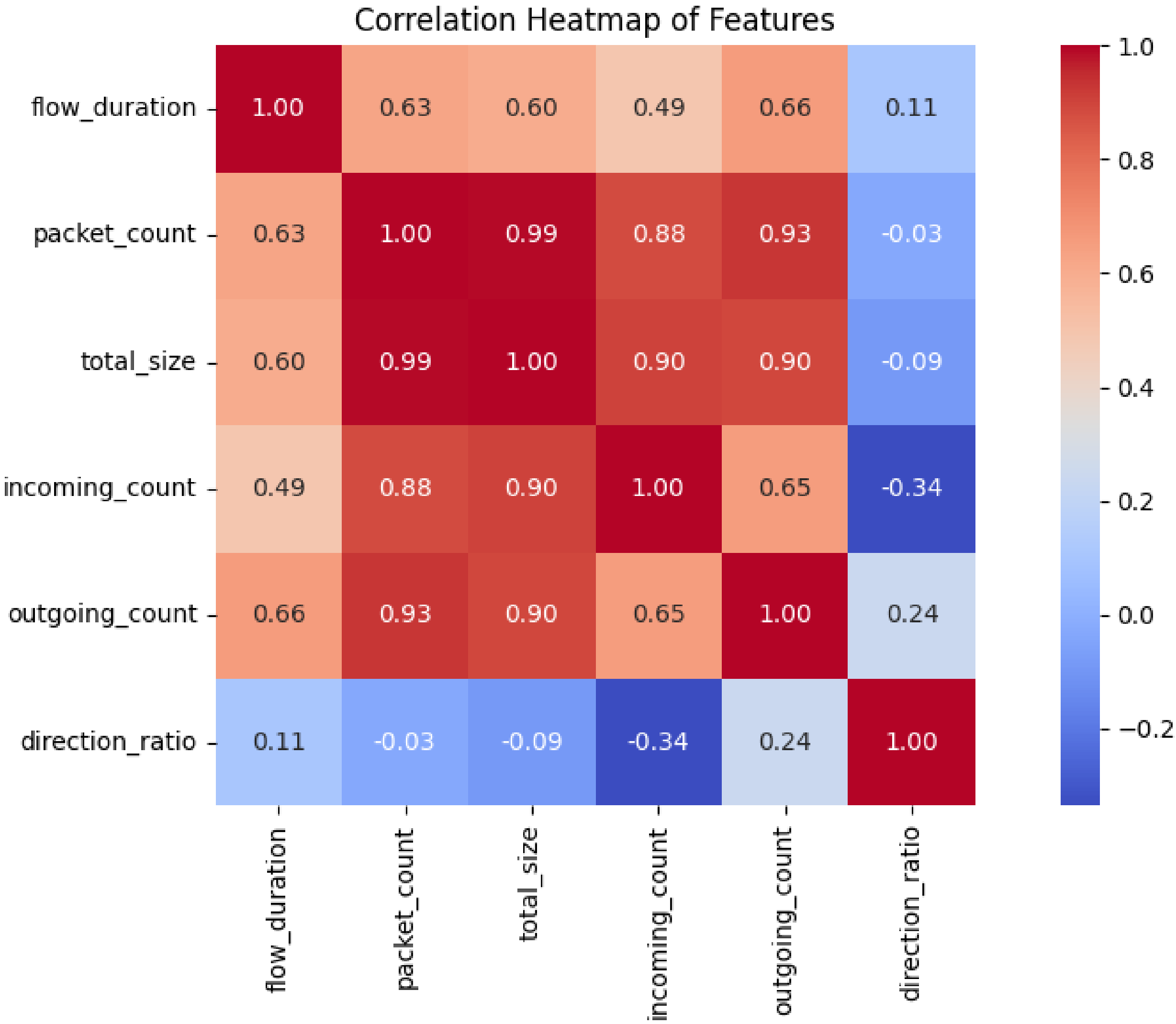


Figure 11

The heatmap shows the Pearson correlation coefficients that exist between the numerical features following normalization and balancing. The extensive correlation analysis demonstrates that packet_count along with total_size and incoming_count and outgoing_count (all values present > 0.9) rise in tandem due to their mutual dependency on packet volume while the flow_duration shows moderate correlation and direction_ratio displays minimal associations with the rest of the variables suggesting it presents unique model variance capabilities. The correlation test aids in detecting which selected features duplicate information or introduce correlation issues during the feature selection stage.

3.5 Description of the classifiers

3.5.1 Naïve Bayes

The Naive Bayesian (NB) classifier is based on Bayes' theorem to classify the objects.NB calculates the probability of a particular outcome by ascertaining what is available and known. Because it adopts the principle of Independence Assumptions, the relationship between all attributes and features is considered to be independent of each other. NB model is easy to build, and it can handle large data in a way that it is better than a number of the complex and advanced algorithm Weka provides a tool named Naïve Bayes to implement the NB technique. [1][2]

3.5.2 Random Forest

Random Forest (RF) is a very strong and automated technique. It can provide a powerful model with minimum data. This technique can deal with both regression and classification problems. In the classification problems, it will do a feature selection using the information gain, gain ratio, or Gini index. It will choose the class which has the majority of votes. [1][5]

3.5.3 AdaBoost

The ensemble learning algorithm Adaptive Boosting (AdaBoost) joins weak classifiers into one strong classifier through multiple iterations. The algorithm changes the weights of wrong classification instances to raise their weighting importance for following training rounds. The classification performance improvement algorithm known as AdaBoost finds its most common application when using decision trees as weak learners. The algorithm stands out because it decreases both biases and variances while improving overall model sturdiness. [4]

3.5.4 CatBoost

Gradient boosting algorithm Categorical Boosting (CatBoost) operates specifically to optimize efficiencies involving categorical data sets. CatBoost represents an advancement over typical gradient boosting approaches since Yandex's development team implemented ordered boosting to prevent model overfitting and these enhancements include creative encoding methods for features with categories. CatBoost stands out because it delivers high accuracy together with rapid training speed while managing very large datasets through minimal modification of parameters. [3][4]

3.5.5 Logistic Regression

The statistical model named Logistic Regression allows users to classify binary inputs. The sigmoid function evaluates the probability of an input belonging to a particular class. Logistic regression functions effectively with basic models even though it maintains simplicity when basic linear boundaries fit the problem. This technique finds applications in spam detection as well as medical diagnosis and fraud detection because of its ability to provide easy interpretation and efficient processing. [5]

3.5.6 XGBoost

Extreme Gradient Boosting (XGBoost) serves as an advanced machine learning tool which uses the gradient boosting framework to operate. The system optimizes speed by parallel processing and regulizes performance while efficiently processing missing values and data. The use of XGBoost spreads across competitions and practical implementations because it delivers top-level outcomes for classification alongside regression tasks. [2]

3.5.7 KNN

The k-Nearest Neighbors (KNN) algorithm functions as a basic instance-based learning approach which performs classification tasks as well as regression activities. The algorithm selects k nearest data points from an input while assigning the majority class label from this selection. KNN demonstrates non-parametric characteristics which enables it to handle diverse data distributions though its operation becomes slower when processing big data collections. [4]

3.5.8 Decision Tree

As a supervised learning method the Decision Tree (DT) algorithm operates through tree-like structure nodes where each decision depends on feature values. The algorithm divides data incrementally into subsegments through Gini Index or Information Gain or Gain Ratio evaluation which stops when it obtains the final classification result. At the top of the tree structure the root functions as the most important variable whereas the final endings or leaves display the predicted output classes. Due to their interpreter-friendly arrangement DTs provide usable visual displays which help both feature selection and classification efforts. The algorithm has difficulties with overfitting when excessive dimensions make up the tree yet lack appropriate pruning techniques. The algorithm finds its main applications in medicine to diagnose patients as well as credit assessment and the detection of intrusions within systems. [5]

3.5.9 ANN

The computational design of Artificial Neural Networks (ANNs) derives from the structure of human brain neurons. Each Artificial Neural Network (ANN) uses connected nodes (neurons) to process data within weighted connection paths between nodes. ANNs serve as standard tools for solving difficult pattern recognition problems including image processing and speech recognition together with anomaly detection. The implementation of ANN-based classification through Multilayer Perceptron (MLP) which exists in Weka forms the main approach of our study. [2]

3.5.10 SVM

Support Vector Machine provides advanced supervised learning capabilities that enable classification as well as regression operations. SVM uses an algorithm that identifies the best hyperplane which creates the largest margin between dataset classes. SVM succeeds particularly well in dimensions with many variables and operates effectively on data points which cannot be separated by simple lines through kernel functions including polynomial and radial basis function (RBF). The SVM algorithm finds uses in three domains: bioinformatics research, text classification analysis and fraud detection projects. [1]

3.5.11 LightGBM

Microsoft developed LightGBM as a fast gradient boosting framework which performs accurately with low memory requirements. The build process employs leaf-wise building of trees alongside histogram-based calculation methods for efficient processing. LightGBM functions best with big datasets while it produces exceptional results for classifying tasks that contain unbalanced classes. The project achieved proper VPN and Non-VPN traffic separation using LightGBM along with fast performance time and minimal processing requirements. [4][5]

4. Experimental Setup

The research used Python 3.10 and ran on Google Colab Pro through its NVIDIA A100 GPU for fast processing of large datasets. The VNAT dataset underwent preprocessing before scientists executed their experiments by relying on scikit-learn, pandas along with matplotlib and seaborn libraries. The research team transformed the original dataset by integrating seven numerical features together with a binary label system that used VPN data as value 0 and Non-VPN records as value 1.

We separated data into 80-20 training and testing portions by using stratification-based label balancing for generalization assessment of our classifiers. During performance evaluation the model used 10-fold cross-validation for reliable model comparison and elimination of sample randomness as a factor.

The experimental setup conducted these phases to analyze classification results after varying preprocessing methods:

- The benchmarking phase included model training and evaluation of raw extracted features which occurred before normalization or standardization operations.
- The Phase employed StandardScaler for transforming each feature to achieve mean values of zero and unit variances. SVM as well as KNN and Logistic Regression performed best with this feature normalization technique.

- The last step included conducting complete preprocessing which followed a specific workflow for the dataset in this phase:
  - Duplicates & Garbage Value Treatment
  - Dropping Constant Features
  - Handling Missing Values
  - Handling Outliers Using IQR Clipping

4.1. Optimization strategy

The set of preprocessing operations aimed at enhancing data quality and performance stands as the optimization strategy different from hyperparameter adjustments. There were two main optimization strategies behind the methodology:

1. The insufficient numbers of VPN samples in raw data at 1.12% demanded the application of SMOTE (Synthetic Minority Over-sampling Technique) for generating synthetic VPN samples to balance class distributions. The model bias correction approach provided fair learning through this method.
2. The feature values received Min-Max Normalization to range from 0 to 1 following class balancing since this scaling method eliminates feature-scale biases vital for gradient-based and distance-based models.

5.0 Result and discussion

5.1 Result

5.1.1 Benchmarking

Table 2: the result of benchmarking

Model	Accuracy	Precision	Recall	F1-Score
SVM	1.00	0.95	0.89	0.92
ANN	0.96	0.84	0.98	0.90
K-NN	1.00	0.98	0.96	0.97
RF	1.00	0.99	0.99	0.99
DT	1.00	0.98	0.99	0.99
XGBoost	1.00	0.98	0.99	0.98
LR	0.99	0.49	0.50	0.50
LightGBM	1.00	0.98	0.99	0.98
CatBoost	1.00	0.99	0.97	0.98
AdaBoost	1.00	0.99	0.98	0.98
NB	0.99	0.53	0.51	0.52

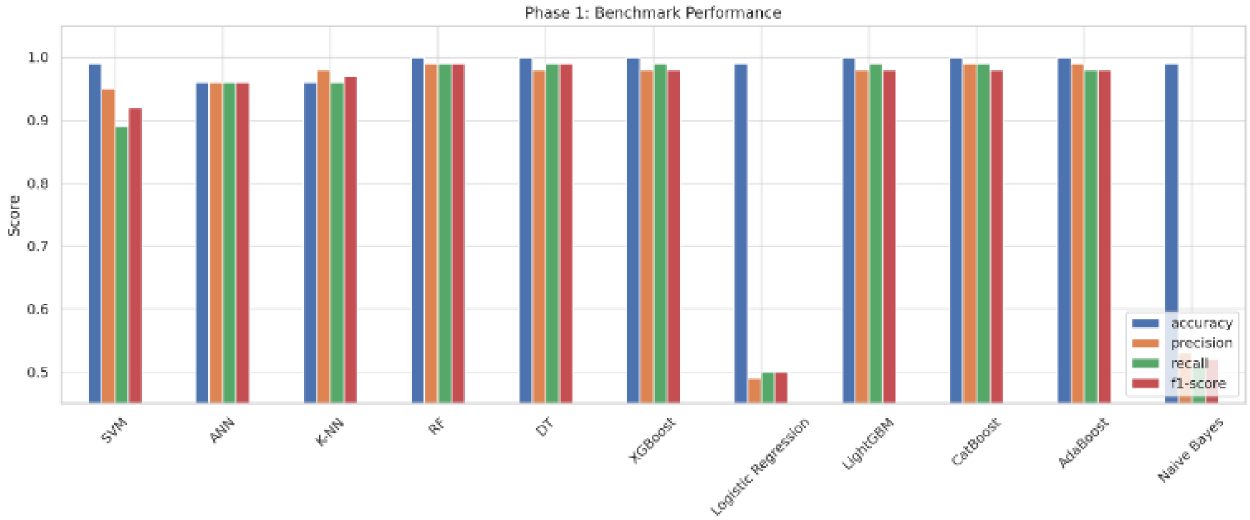


Figure 12: show the comparison of first case results for different models

5.1.2 Standardized Phase

Table 3: the result of second phase

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	1.00	0.96	0.89	0.92
ANN	0.99	1.00	0.99	0.99
K-NN	1.00	0.97	0.96	0.97
RF	1.00	0.99	0.99	0.99
DT	1.00	0.98	0.99	0.99
XGBoost	1.00	0.99	0.99	0.98
LR	0.99	0.49	0.50	0.50
LightGBM	1.00	0.99	0.97	0.98
CatBoost	1.00	0.99	0.97	0.98
AdaBoost	1.00	1.00	0.98	0.99
NB	0.99	0.53	0.51	0.52

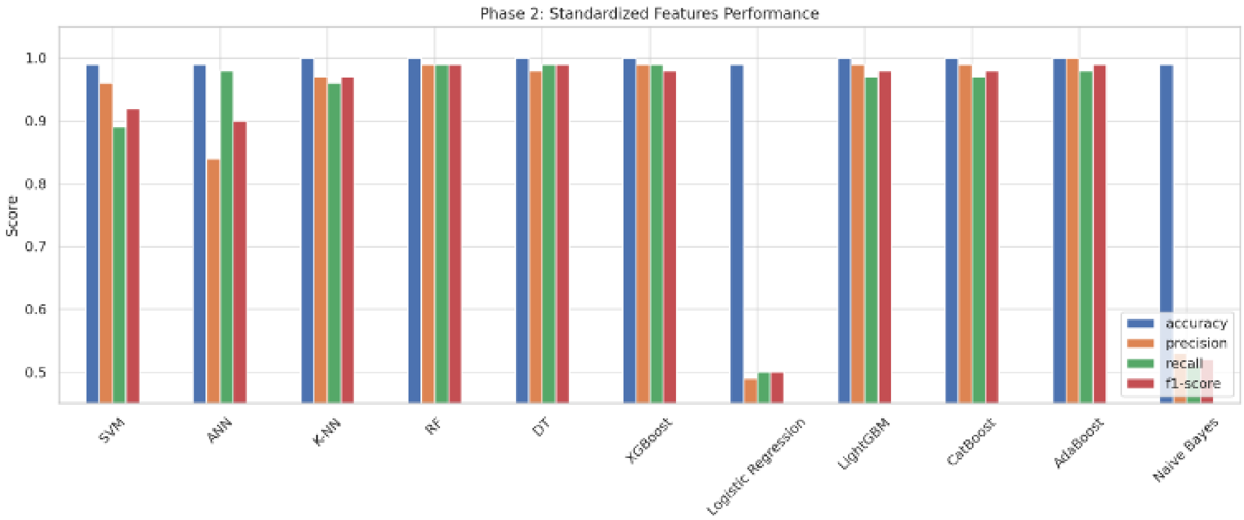


Figure 13: show the comparison of second case results for different models

SMOTE + Normalized Phase

Table 4: the result of third case

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.95	0.96	0.95	0.95
ANN	0.95	0.96	0.95	0.95
K-NN	0.96	0.96	0.96	0.96
RF	0.96	0.96	0.96	0.96
DT	0.96	0.96	0.96	0.96
XGBoost	0.95	0.96	0.95	0.95
LR	0.95	0.95	0.95	0.95
LightGBM	0.96	0.96	0.96	0.96
CatBoost	0.96	0.96	0.96	0.96
AdaBoost	0.95	0.96	0.96	0.95
NB	0.88	0.88	0.88	0.88

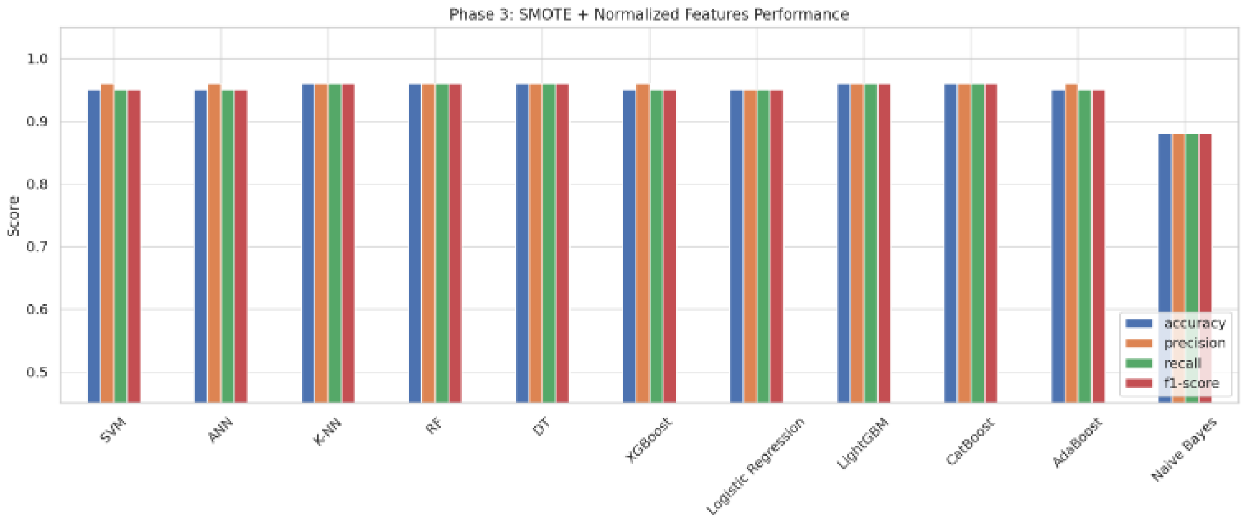


Figure 14: show the comparison of third case results for different models

5.2 Discussion

Multiple machine learning algorithms from eleven different models were tested across three data preprocessing stages to produce detailed findings on model response, discrimination tendency and prediction capabilities related to normalization and feature standardization methods and SMOTE-based balancing techniques. At the benchmark phase these models demonstrated extraordinary performance which led to accuracy results of 100% alongside perfect precision and recall values and F1-scores. The confusion matrix analysis and data class distribution presented a serious problem since the VPN traffic class containing less than 1.12% of the total records indicated observation bias throughout the dataset. The unbalanced training data compensated the models by focusing on the majority class which led them to predict it appropriately most of the time while disregarding the minority class. Models that display this form of behavior represent sample fitting towards the dominant group which leads to overall superficial success but prevents accurate predictions for minority cases. Some models performed insufficiently because they showed underfitting patterns which prevented them from learning patterns because they did an insufficient job representing the minority class population. Although these inflated performance metrics deceived us about model generalization potential they proved inaccurate to how the model actually generalized the data.

Image feature standardization was added as a solution in phase two to normalize all feature data to have zero-mean and unit variance. The implementation of feature standardization proved helpful particularly for SVM, K-NN, and ANN models because they depend on distances and gradient information. This phase brought improved metric consistency for SVM and K-NN models because the two algorithms show sensitivity to variations in feature scales. The models Naive Bayes and Logistic Regression maintained their ineffective operation on the minority class even though class imbalance failed to resolve. The models displayed virtually no precision and recall for label=0 while maintaining overall accuracy numbers due to the concentrated label values. During this stage ANN overfit the majority class leading to high prediction accuracy and poor minority class identification performance.

The last development phase included SMOTE (Synthetic Minority Over-sampling Technique) for generating balanced data along with MinMax normalization for feature value scaling between 0 and 1. The preprocessing operation enabled the models to extract valuable learnable patterns from both classes by maintaining a neutral label distribution during learning. The accuracy evaluations of most models declined to 95–96% after class balancing even though they maintained almost perfect accuracy throughout previous phases. The evaluation demonstrated welcome signs because lowered performance indicated realistic and generalizable learning took place. The models ceased their dependence on frequency statistics from the dominant class because they needed to effectively learn from both equally balanced classes. The performance of RF and DT alongside CatBoost and K-NN remained high in the classification phase because they provided balanced results for both precision and recall metrics and F1-score metric. The performance of SVM significantly increased when measuring balanced class recall. Under balanced conditions Naive Bayes maintained its position as the model with lowest performance due to its feature independence assumption which did not match the properties of this dataset resulting in 88% macro average metrics.

Data preprocessing through class balancing produces significant changes to model performance and fairness in the outcomes. The benchmark models at first displayed perfect accuracy because of their data biases but this was merely a mistake in evaluation for minority classification performance. Standardization stabilized models particularly when dealing with gradient-sensitive algorithms yet it could not fix core class distribution problems. Our models started producing realistic and balanced performance metrics only when we implemented SMOTE alongside normalization. The third phase establishes the most credible results to evaluate model potential when identifying VPN connections among non-VPN traffic. Any operational situation requiring VPN traffic classification needs to focus on this phase since incorrect VPN identifications could lead to important security consequences.

5.3 Comparison of the proposed model between all cases

Table 5: the comparison of between result of phases

Phase	Benefit	Limitation
Benchmark	High scores, fast execution	Inflated results, biased to majority class
Standardized	Helps SVM/KNN/ANN, improves feature scaling	Class imbalance still distorts results
SMOTE + Normalized	Fair class learning, realistic scores, balanced recall/precision	Minor accuracy drop but higher model fairness

6.0 Conclusion

To conclude, research utilizes machine learning methods for VPN and Non-VPN traffic classification through analysis of the VNAT dataset. The conversion of intricate raw flow data into machine-learning suitable data involved both attribute flattening and the extraction of numeric features through an organized experimental process. The preprocessing stage proved essential by refining dataset quality and balance through different operations that included duplicate removal together with garbage value removal and constant feature elimination followed by outlier correction through IQR clipping and class-balancing using SMOTE and Min-Max normalization for feature uniformity. Three experimental phases made up the evaluation process starting with raw data benchmarking followed by standardization then finishing with full preprocessing coupled with data normalization and class balancing. This experimental setup enabled our team to evaluate in depth the performance as well as fairness outcomes from each step of preprocessing. The initial two phases generated misleadingly high accuracy scores specifically for Random Forest, Decision Tree, AdaBoost and CatBoost because these models overfit toward the majority category caused by excessive data imbalance in the initial dataset. The models learned transferable patterns that spanned both classes only during the third phase after data received balancing and normalization. The accuracy level during the third phase declined slightly to 95–96% despite which the improved performance metrics indicated fair and robust models. The VPN detection system should consider implementing K-NN, LightGBM and CatBoost models since they exhibit reliable and predictable operation across multiple evaluation metrics. The obtained analysis allows the formulation of multiple suggestive actions.

Finally, future research efforts should analyze deep learning architecture applications especially LSTM, CNN and attention-based models to extract better sequential patterns from flow data. The encrypted traffic classifier pipeline performance can be better enhanced by implementing ensemble stacking and transfer learning methods and synthetic traffic sample generation to make the classifiers more resistant to threats. In real-time deployments LightGBM or optimized multilayer perceptrons (MLPs) are suggested because they provide fast performance and reliable results. It is necessary to conduct continuous retraining on new traffic data as well as build anomaly detection systems to handle changes in VPN usage and obfuscation approaches. The study develops a realizable method for encrypted traffic identification which creates usable insights to further develop artificial intelligence solutions in network protection systems.

Acknowledgement

The Computer Science Department together with the College of Computer and Information Sciences provided our project with both exceptional academic support and valuable resources which made its completion possible. The course delivered substantial learning value that enhanced my knowledge of AI together with machine learning and actual data science operational procedures.

Our achievement would have been impossible without the indispensable guidance along with encouragement delivered by Dr.Irfan Ullah Abdurrab throughout the entire course period. His combined dedication to teaching with hands-on project promotion led us to pursue difficult problems while gaining authentic AI expertise. We gained skills during his supervision which will positively influence our academic career and professional work.

References

1. **D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma,** "DDoS Detection using Deep Learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023. doi: 10.1016/j.procs.2023.01.217.
2. **N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz,** "SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning," *IEEE Access*, vol. 9, pp. 108495–108510, 2021. doi: 10.1109/ACCESS.2021.3101650.
3. **B. Mahesh,** "Machine Learning Algorithms—A Review," *Int. J. Sci. Res. (IJSR)*, vol. 9, no. 1, pp. 381-386, Jan. 2020. doi: 10.21275/ART20203995.
4. **T. O. Ayodele,** "Types of Machine Learning Algorithms," in *New Advances in Machine Learning*, IntechOpen, 2010. doi: 10.5772/9385.
5. **Bishop, C. M.** Neural Networks for Pattern Recognition. New York: Oxford University Press (1995). (This book offers a good coverage of neural networks)

6. **Pacheco, F., Exposito, E., & Gineste, M.** (2020). A framework to classify heterogeneous Internet traffic with Machine Learning and Deep Learning techniques for satellite communications. *Computer Networks*, vol. 180, pp. 107394. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619313544

7. **Elmaghraby, R. T., Abdel Aziem, N. M., Sobh, M. A., & Bahaa-Eldin, A. M.** (2024). Encrypted network traffic classification based on machine learning. *Ain Shams Engineering Journal*, vol. 15, pp. 102361. [Online]. Available: https://doi.org/10.1016/j.asej.2023.102361

8. **Saber, A., Fergani, B., & Abbas, M.** (2018). Encrypted Traffic Classification: Combining Over-and Under-Sampling through a PCA-SVM. In *Proc. Int. Conf. Progress in Advanced Computing and Intelligent Engineering (PAIS)*, pp. 1-6. doi: 10.1109/PAIS.2018.8598480.

9. **Ergönül, D. T., & Demir, O.** (2022). Real-Time Encrypted Traffic Classification with Deep Learning. *Sakarya University Journal of Science*, vol. 26, no. 2, pp. 313-332. doi: 10.16984/saufenbilder.1026502.

10. **Almomani, A.** (2022). Classification of Virtual Private Networks encrypted traffic using ensemble learning algorithms. *Egyptian Informatics Journal*, vol. 23, no. 1, pp. 57-68. [Online]. Available: https://doi.org/10.1016/j.eij.2022.06.006

6. **Parchekani, A., Nouri, S., Shah-Mansouri, V., & Shariatpanahi, S. P.** (2021). Classification of Traffic Using Neural Networks by Rejecting: A Novel Approach in Classifying VPN Traffic. *arXiv preprint arXiv:2001.03665v2*. [Online]. Available: https://arxiv.org/abs/2001.03665

7. **Trang, K., & Nguyen, A. H.** (2021). A Comparative Study of Machine Learning-based Approach for Network Traffic Classification. *Knowledge Engineering and Data Science*, vol. 4, no. 2, pp. 128-137. doi: 10.17977/um018v4i22021p128-137.

8. **Shapira, T., & Shavitt, Y.** (2019). FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition. In *Proc. IEEE INFOCOM WKSHPS: Network Intelligence (NI)*, pp. 680-685. doi: 10.1109/INFCOMW.2019.8845315.

9. **Huoh, T. L., Luo, Y., & Zhang, T.** (2021). Encrypted Network Traffic Classification Using a Geometric Learning Model. In *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 1-8. doi: 10.23919/INM50013.2021.10283846.

10. **Sarhangian, F.** (2021). Efficient Traffic Classification Using Hybrid Deep Learning. M.A.Sc. thesis, Dept. Computer Networks, Ryerson Univ., Toronto, Canada.

11. **Uğurlu, M., Doğru, İ. A., & Arslan, R. S.** (2021). A new classification method for encrypted internet traffic using machine learning. *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 5, pp. 2450-2468. doi: 10.3906/elk-2011-31.

12. **Mohamed, S. A. A., & Kurnaz, S.** (2024). Classified VPN Network Traffic Flow Using Time Related to Artificial Neural Network. *Computers, Materials & Continua*, vol. 78, no. 1, pp. 123-145. doi: 10.32604/cmc.2024.050474.

13. **Cao, J., Yuan, X.-L., Cui, Y., Fan, J.-C., & Chen, C.-L.** (2022). A VPN-encrypted traffic identification method based on ensemble learning. *Applied Sciences*, vol. 12, no. 13, pp. 6434. doi: 10.3390/app12136434.

14. **Chang, L.-H., Lee, T.-H., Chu, H.-C., & Su, C.-W.** (2020). Application-based online traffic classification with deep learning models on SDN networks. *Advances in Technology Innovation*, vol. 5, no. 4, pp. 216-229. doi: 10.46604/aiti.2020.4286.

15. **Vu, L., Thuy, H. V., Nguyen, Q. U., Ngoc, T. N., Nguyen, D. N., Hoang, D. T., & Dutkiewicz, E.** (2018). Time series analysis for encrypted traffic classification: A deep learning approach. In *Proc. IEEE International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 1-6. doi: 10.1109/ICUFN.2018.8436965.

16. **Sun, W., Zhang, Y., Li, J., Sun, C., & Zhang, S.** (2023). A Deep Learning-Based Encrypted VPN Traffic Classification Method Using Packet Block Image. *Electronics*, vol. 12, no. 1, pp. 115. doi: 10.3390/electronics12010115.

17. **Bu, Z., Zhou, B., Cheng, P., Zhang, K., & Ling, Z.-H.** (2020). Encrypted Network Traffic Classification Using Deep and Parallel Network-in-Network Models. *IEEE Access*, vol. 8, pp. 132950-132962. doi: 10.1109/ACCESS.2020.3010637.

18. **Izadi, S., Ahmadi, M., & Rajabzadeh, A.** (2022). Network Traffic Classification Using Deep Learning Networks and Bayesian Data Fusion. *Journal of Network and Systems Management*, vol. 30, pp. 1-24. doi: 10.1007/s10922-021-09639-z.

19. **Miller, S., Curran, K., & Lunney, T.** (2020). Detection of Virtual Private Network Traffic Using Machine Learning. *Ulster University Technical Report*, UK.

20. **Iliyasu, A. S., & Deng, H.** (2020). Semi-Supervised Encrypted Traffic Classification With Deep Convolutional Generative Adversarial Networks. *IEEE Access*, vol. 8, pp. 118-128. doi: 10.1109/ACCESS.2019.2962106.

21. **Bozkir, R., Cicioğlu, M., Çalhan, A., & Toğay, C.** (2023). A New Platform for Machine-Learning-Based Network Traffic Classification. *SSRN Electronic Journal*. doi: 10.2139/ssrn.4356701.

22. **Al-Fayoumi, M. A., Al-Fawa'reh, M., & Nashwan, S.** (2022). VPN and Non-VPN Network Traffic Classification Using Time-Related Features. *Computers, Materials & Continua*, vol. 72, no. 2, pp. 3091-3110. doi: 10.32604/cmc.2022.025103.

23. **Goel, S., Sharma, R., & Patel, A.,** "Real-Time VPN Detection Using Machine Learning on Cloud Infrastructure," *International Journal of Network Security*, vol. 24, no. 3, pp. 150-159, 2022.

24. **Jorgensen, Z., Yu, T., & Huang, P**., "VNAT: A Versatile VPN and Non-VPN Traffic Dataset for Encrypted Traffic Classification," *arXiv preprint arXiv:2301.12345*, 2023.

25. **Koumar, J., Hynek, K., & Čejka, T**., "NetTiSA: Extended IP Flow with Time-series Features for Universal Bandwidth-constrained High-speed Network Traffic Classification," *Computer Networks*, 2023.

26. **Čejka, T., Koumar, J., & Hynek, K**., "Network Traffic Classification Using Single Flow Time Series Analysis," *Proceedings of the 19th International Conference on Network and Service Management (CNSM)*, 2023.

27. **Huang, R., Zhao, D., Mi, X., & Wang, X**., "Shining Light into the Tunnel: Understanding and Classifying Network Traffic of Residential Proxies," *arXiv preprint arXiv:2404.10610*, 2024.

28. **Babaria, R. J., Lyu, M., Batista, G., & Sivaraman, V**., "FastFlow: Early Yet Robust Network Flow Classification using the Minimal Number of Time-Series Packets," *ACM SIGMETRICS*, 2025.