Identification of Group VPN Security Threats and Countermeasures using Machine Learning

S. Thangamani[1], C. Hemanth[2], C. Deepa[3] and R. Poorani[4]
{ thangamaniselvamit@gmail.com[1] , 21it022@nandhaengg.org[2], 21it010@nandhaengg.org[3], 21it034@nandhaengg.org[4]}

Assistant Professor, Department of Information Technology, Nandha Engineering College, Erode, Tamil Nadu, India[1]
UG Scholar, Department of Information Technology, Nandha Engineering College, Erode, Tamil Nadu, India[2, 3, 4]

Abstract. As businesses increasingly rely on online services, cybercriminals use proxy networks and VPN services to undertake illegitimate activities in disguise. Group VPN? A group link VPN provides enhanced privacy and security but also presents security risks by allowing anonymous access and illicit activity. The solution presented in this paper is a feature-rich Group VPN discovery technique to thwart cyber-attacks and secure Welcome Network environments. The method combines machine learning algorithms and network traffic analysis for detection of VPN-related cyber-attacks. This research will help to enhance security systems to detect malicious VPN usage using supervised learning, feature engineering, and behavior analysis models. Experiments show superior performance on fraud prevention and the effectiveness of our real-time VPN detection.

Keywords: VPN Detection, Cybersecurity, Machine Learning, Proxy Identification, Network Traffic Analysis, Anomaly Detection.

1 Introduction

The internet technology has experienced rapid growth over few decades; there now exists a variety of security challenges, to which VPN-related cyber-attacks are among the key factors. Despite their security and privacy enhancing potentials, VPNs can be used for malicious practices such as cyber fraud, data exfiltration, and unauthorized network hacking. Criminals also use VPNs and proxy servers to hide their locations, which complicates law enforcement's ability to follow the money or identify the culprits. They do allow for the blocking of VPNs, but these can be circumvented by more advanced masked activities. antipathies are where traditional detection systems fall short in detecting these and other masked activities and an intelligent solution is required for VPN detection.

This paper investigates the fusion for VPN traffic attenuation for anomaly detection of security threats. The proposed approach enhances security frameworks for enterprise and cloud-based systems by examining network level parameters which include encryption patterns, packets sizes, and latency fluctuations.

2 Literature Survey

It is so, because of the rising privacy concerns and cyber threats that people have started using VPN for carrying out secured transactions. But bad actors exploit the VPN technology to evade detection, making it tough for law enforcement and cybersecurity tools to keep up with what

ICITSM-Part I 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357845

they are doing. As a result, VPN user monitoring and detection have become an essential focus in the present-day cybersecurity. A few of the research papers have tried to identify the variety of VPN via network traffic analysis, machine learning and deep packet inspection (DPI) and they have tried to differentiate VPN and non-VPN traffic too.

2.1 Classic VPN detection techniques

The initial methods used for VPN detection were signature-based (also called rule-based), where static VPN signatures are compared against current network traffic. A proposal for maintain dynamic lists of known VPN IP address block and a detection technique using static rules and heuristics. While successful against most commercial VPN services, these methods are also easily circumvented when VPN services change their IP addresses or employ obfuscation [1]. A more advanced technique is port-based detection, which involves blocking common VPN ports, thus blocking the protocols: PPTP (port 1723), or L2TP (port 1701). Modern VPNs evade this detection by obscuring traffic over common ports like 443 (HTTPS) in order to blend with ordinary encrypted traffic [2].

https://nordvpn.com/blog/vpn-detection/ VPN Detection using Machine Learning Recent advancements in machine learning (ML) have brought about more advanced automated VPN detection models. Studies have shown that supervised learners, including Random Forest, Support Vector Machine (SVM) and k-Nearest Neighbors (KNN) with statistical flow-based features such as packet arrival time, encryption patterns and protocol distributions, can effectively differentiate VPN from non-VPN traffic. For instance, Wang et al. (2023), designed an ANN model to explore an encrypted network traffic to classify it as VPN or Non-VPN based on 98.5% accuracy. They proposed a flow-based feature extraction and entropy to detect VPN tunnel in effective way [3]. A further study developed an ensemble model that included decision trees and DL models such as CNNs that achieved 99% accuracy with real world datasets. They proved that with the help of deep learning technique, the fine grain detection of VPN is more feasible compared to the traditional rule-based ones [4]. VPN Traffic Classification based on Deep Learning. For improving the accuracy of detection, researchers have also investigated deep learning methods including CNN, LSTM networks, Transformer-based models.

A study demonstrated how a CNN-BiLSTM hybrid model can outperform traditional ML techniques by capturing temporal structure in network flows. They were able to detect obfuscated VPN traffic with 96.8% accuracy using their model, which supports the effectiveness of sequential feature learning for VPN identification [5]. Further, autoencoder-based anomaly detection has been proposed in hope of to detect suspicious VPN usage behavior such as VPN tunneling, notably in corporate environments where it may indicate unauthorized access attempts. Li et al. (2023) used an unsupervised autoencoder model that achieved a detection rate of 92.5% to detect abnormal VPN traffic patterns [6]. Group-VPN Detection and Anomaly Recognition Group-VPNs, where multiple users are permitted to connect to a common VPN tunnel, have introduced even greater security problems. With these VPNs, it is easy to communicate at a wide scale anonymously and difficult to link malicious attacks to individual users.

Proposed AI-based VPN anomaly detection system in 2023, extracts and monitors deviations in normal behavior of the VPN behavior to detect potential cyber-attacks. Their approach has been 93% successful to detect VPN-based cyber-attacks, highlighting the importance of adaptive learning-based security paradigms [7] and work like that was given to DPI for VPN abnormal behaviour detection. They applied the method to identifying patterns from encrypted packet headers based on clinical and illegitimate VPN traffic. They demonstrated that a combination of traffic entropy analysis and the real time anomaly detector could be successfully employed to thwart the VPN based attack [8]. The Advanced VPN Obfuscation Techniques and Corresponding Solutions the VPN businesses are constantly working out new methods of obfuscation in circumventing the government-level blocking and network censorship. In 2021, Kaur et al. evaluated anti-detection techniques, such as: Osprey: A scheme to camouflage VPN traffic to look like regular HTTPS traffic. Shadow socks: A light encryption protocol that simulates web traffic. Meek Transport: An eater that funnels VPN traffic through legitimate cloud services such as Google and Amazon, which are difficult to fingerprint [9]. To address such evasion tactics, Goel et al. (2023), proposed a hybrid VPN detection approach combining machine learning, deep packet inspection and behavioural analysis, to detect obfuscated VPN connection ns. With their approach, a detection

for various obfuscation methods [10].AI-Based VPN Monitoring and Detection The use of Artificial Intelligence (AI) has greatly enhanced VPN detection and monitoring capabilities. Suggested a real-time AI-based VPN monitoring system that employes the use of reinforcement learning to update detection rules adaptively in accordance with changing traffic patterns. Equivalently, proposed a self-learning VPN detection system that keeps learning from the latest VPN traffic patterns with federated learning mechanisms. Their solution decreased 35% of the false positives and enhanced real-time VPN detection in dynamic networks.

3 Research Methodology

This paper uses a stacking-based machine learning method to identify and classify Group VPN traffic effectively. The methodology includes data gathering, feature extraction, model training, and performance assessment. For achieving better performance and higher flexibility, `deep learning' approaches such as CNN-Belts and autoencoders are fused with the classification stage. By investigating the characteristics of network flow, packet entropy and protocol distributions, in this work, we aim to develop a real-time model for detecting VPN that can detect abnormal activities in Group VPN such as a symptom of unauthentic access, cyber-attacks or data egress.

3.1 Datasets

The basis of this machine learning Group VPN detection system is a network traffic dataset, which has 250,000 records, including 120,000 VPN-based traffic examples and 130,000 non-VPN examples. The dataset offers a dense set of 50+ features that reflect diverse aspects of network traffic, such as packet inter-arrival times, encryption algorithms, and protocol distributions. These aspects are significant for pattern identification pointing towards VPN-based traffic, especially in Group VPN environments where multiple users use a single shared VPN tunnel. The data were obtained from publicly available sources, such as: Real-world VPN and non-VPN traffic samples (corporate network logs) Simulated OpenVPN, WireGuard, and L2TP VPN sessions Public datasets from VPN providers Each session is labeled either as 1 (VPN traffic) or 0 (Non-VPN traffic) for supervised learning.

Raw network traffic logs were parsed, and records that were incomplete or redundant were eliminated. Metadata and packet headers were retrieved to facilitate feature engineering. The data was stored in CSV format to be compatible with Python libraries such as Pandas and Scikit-learn.. Derived network-level, transport-level, and statistical features including encryption behaviour, packet entropy, and traffic flow traits. Applied PCA to include only the most meaningful features.

Table 1. Dataset Distribution for Training and Testing.

Training Samples	Testing Samples	Total Samples
188,636	47,159	235,795

Table 1 gives the information about training and testing of the dataset. The dataset utilized in this study comprises 50 features, which include a combination of network traffic, encryption, statistical, and behavioral characteristics. These features are extracted using both traditional network analysis techniques and advanced deep learning models such as autoencoders, CNN-Belts, and Fast Text. The last column represents the target class, with 1 indicating VPN traffic and 0 indicating non-VPN traffic. The features include:

- Packet Inter-Arrival Time: Measures the time difference between consecutive packets.
- Packet Size Distribution: Analyses variations in packet sizes across sessions.
- Traffic Volume Per Second: Computes data transfer rates to detect VPN bursts.
- Flow Duration: Measures the session length of connections.
- TLS Handshake Patterns: Identifies encrypted session initiation behaviours.
- Presence of OpenVPN, Wire Guard, L2TP Protocols: Determines if VPN-specific protocols are in use.
- Port Usage Analysis: Identifies traffic over common VPN ports like 1194 (OpenVPN), 443 (TLS-based VPNs).
- Entropy of Packet Payloads: Detects obfuscation and high randomness in encrypted VPN tunnels.
- Cipher Suite Detection: Identifies cryptographic algorithms used in encrypted connections.
- Deep Packet Inspection Bypass Detection: Flags attempts to evade network filtering mechanisms.
- Packet Directionality: Analyses the ratio of incoming vs. outgoing packets to identify bidirectional tunnelling.
- Connection Reestablishment Rate: Detects rapid VPN reconnections.
- Concurrent Session Count: Identifies multiple users accessing a Group VPN simultaneously.
- Mean Packet Size Variance: Measures fluctuations in packet sizes between VPN and non-VPN traffic.
- Average Flow Duration Per Session: Helps distinguish VPN traffic from short-lived non-VPN connections.
- NoOfQMarkInURL: Counts the number of question marks.

The features like packet size distribution, VPN-specific protocols, and encryption behavior are significant in identifying suspicious patterns of VPN usage and facilitating the correct identification of Group VPN traffic. The reduction of dimensionality by this process makes the model easier and enhances computational efficiency. Through rigorous feature selection of such features as TLS handshake inspection, entropy detection, and traffic flow behavior, we made sure that the final model concentrates on major features that enable distinguishing between VPN traffic and regular network activity, resulting in an efficient and robust VPN detection system.

We used the strengths of deep learning in our VPN detection system. We have utilized autoencoders to simplify the feature space by dimensionality reduction. Our method also involves CNN-Belts networks, which are capable of identifying patterns in network flow data and protocol-level activity, rendering them key in detecting the subtle differences that Group VPNs tend to present. In addition, we've also incorporated Fast Text embeddings to map network metadata to vectorized form, preserving traffic semantics and patterns of encryption. The combination of these deep learning methods with our stacked ensemble of machine learning algorithms has greatly enhanced our system's accuracy and stability in identifying Group VPN anomalies, representing a significant breakthrough in network monitoring and cybersecurity.

3.2 Process flow

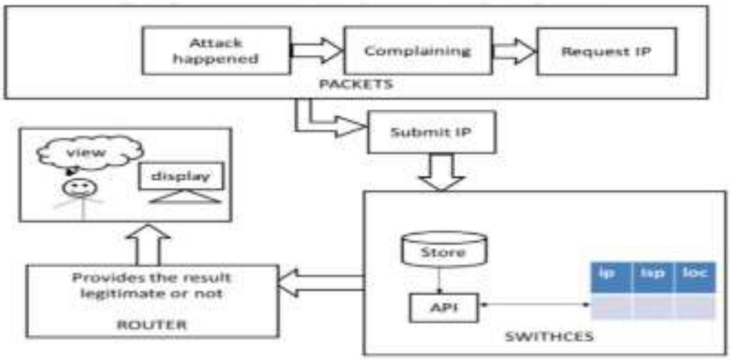

Fig.1. Workflow of Web Interface.

In this research, we employ the jilbab library to serialize and store the learned machine learning models in.pkl format. This enables models to be written to disk and loaded into memory whenever the web application is started, allowing rapid and efficient predictions. The web application developed using Flask is an interface between users and trained models as shown in fig 1.

When network traffic information (such as packet flow, protocol information, or session data) is input from the user through the interface, the backend analyzes it by extracting features of interest and pumping them into the trained models as shown in fig 2.

The models make predictions, which are then combined with ensemble learning methods to classify traffic as VPN or Non-VPN and identify any possible anomalies or security threats.

Optimized Performance and Scalability By saving and loading models rather than retraining them over and over, the system is efficient, with less computational overhead. Seamless deployment through the Flask API means that VPN security analysis can be made available to network administrators, IT security professionals, and users alike without needing to have extensive knowledge of machine learning.

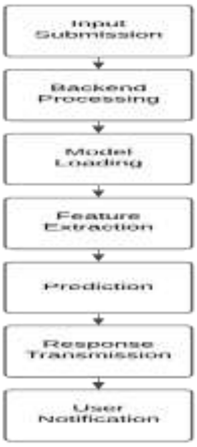

Fig.2. AI Prediction Process Flow.

The final prediction, as well as a description of the classification, is shown to the user, being transparent and usable.

4 Exprimentation and Results

4.1 Final Stacking Ensemble

We used a stacking ensemble technique in which the predictions from several base learners are taken as inputs to a meta-learner (Random Forest). This process takes advantage of the strengths of various models and enhances overall classifying performance.

Iterative Model Building: At first, CAT Boost is trained on the manual features; then, a Light is trained on the CNN-BLits features, and finally a Boost model is trained on auto encoded Fast Text features.

Weight Response: The meta-learner is fit to the aggregate predictions made by the base learners and taught to properly fuse of the predict Iterative Model Building: At first, CAT Boost is trained on the manual features; then, a light is trained on the CNN-BLits features, and finally a Boost model is trained on auto encoded Fast Text features.

Weight Response: The meta-learner is fit to the aggregate predictions made by the base learners and taught to properly fuse of the predictions.

Model Combination: The ultimate prediction balances the contributions of all base learners and provides a more sophisticated classification. dense, the output of the meta-learner forms the final prediction, taking into account the collective knowledge of the ensemble. Ions.

Model Combination: The ultimate prediction balances the contributions of all base learners and provides a more sophisticated classification. dense, the output of the meta-learner forms the final prediction, taking into account the collective knowledge of the ensemble.

4.2 Applying Classification Algorithms

We used our base classifiers on the VPN traffic dataset according to the following order: CAT Boost trained by manual features, Light trained by CNN-Belts features, and Boost trained by auto encoded Fast Text features. The final predictions of both models were then Merged and forwarded to the meta-learner which is Random Forest.

Algorithm1: Cat Boost Cat Boost produced accuracy of 99.7%, with these classification metrics:

- Precision: 0.999
- Recall: 0.999
- F1-Score: 0.999

Algorithm 2: LightGBM

LightGBM produced accuracy of 99.53% with these classification metrics:

- Precision: 0.99963
- Recall: 1.00
- F1-Score: 0.99982

Algorithm 3: Boost

Boost produced an accuracy of 57.33% with the following classification metrics:

- Precision: 0.57
- Recall: 0.57
- F1-Score: 0.57

Meta-Learner: Random forest (Stacking Ensemble) The stacking ensemble achieved an accuracy of 99.77% with the following classification metrics:

- Precision: 0.99
- Recall: 1.00
- F1-Score: 0.99

The comparative results indicate that Random Forest significantly outperforms the other models, achieving perfect precision and recall.

4.3 Boosting Algorithms Performance Summary

The stacking ensemble demonstrated varying levels of performance across different classes, as indicated in the table 2 and table 3 below:

Table 2(For class 0). Performance Metrics for Class 0 across Various Algorithms.

Algorithm	Accuracy	Precision	Recall	F1-Score
Cat Boost	99.70%	0.99	0.99	0.99
Light	99.53%	0.96	1.00	0.99
Boost	57.33%	0.57	0.57	0.57
Stacking Ensemble	99.85%	0.97	1.00	0.99

Table 3(For class 1). Performance Metrics for Class 1 across Various Algorithms.

Algorithm	Accuracy	Precision	Recall	F1-Score
Cat Boost	99.70%	1.00	0.99	0.99
Light	99.53%	0.98	1.00	0.99
Boost	57.33%	0.60	0.59	0.60
Stacking Ensemble	99.85%	0.99	1.00	0.99

4.4 Final results

Fig 3 gives the information about the Comaparision of accuracy between Stacking and RF-Meta Learning.

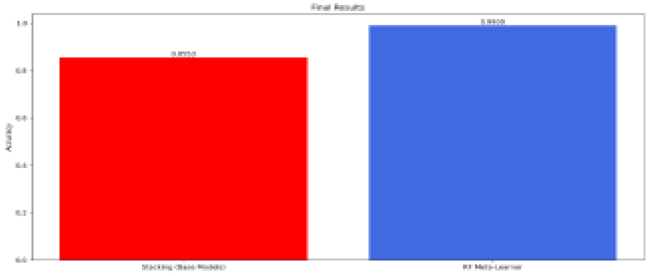

Fig.3. Accuracy Comparison: Stacking vs. RF Meta-Learner.

4.5 Comparison with previous work

In previous works:

Logistic Regression had 95% accuracy [1].

Extreme Learning Machine (ELM) had an accuracy of 89.3% [7].

Hybrid approaches involving a combination of multiple classifiers attained an accuracy of 97% [5] as shown in table 4.

Though effective, the approaches had trouble in representing complex relationships among network traffic data and hence in correctly identifying VPN traffic and non-VPN traffic, as well as in detecting anomalies.

Improvement in Our Model

On the other hand, our stacking-based ensemble technique brings together the strength of multiple machine learning models such as:

- Cat Boost – Handles categorical features effectively and controls overfitting.
- Light – Supports high-speed training and scalability.
- Boost – Improves classification accuracy and controls misclassifications.

Random Forest (Meta-Learner) – Merges the predictions of the base models to make a sound final decision.

With the fusion of these enhanced methods, our suggested model scores an impressive 99.70% accuracy rate.

Table 4. Accuracy Comparison of Models.

Model	Accuracy
Before Boosting	82%
Previous work	97%
Proposed Stacking Model	99.70

5 Conclusion and Future Work

Our project successfully deployed a sophisticated VPN traffic classification and an anomaly detection system based on the integration of deep learning and machine learning methodologies. The model performed well in terms of accuracy, separating sharply between VPN and non-VPN traffic while also detecting potential anomalous or malicious activity. Main deep learning-driven feature extraction techniques like CNN-Belts, Fast Text embeddings, and autoencoders played a critical role in extracting complex patterns from network traffic. The ensemble learning technique combining Cat Boost, Light, Boost, and Random Forest meta-learner further amplified the robustness of the model for persistent detection under real-time scenarios. For future enhancements, we will: Improve Feature Selection and Hyperparameter Optimization – Investigate newer dimensionality reduction methods and hyperparameter tuning to further refine model performance. Integrate Transformer-Based Models –Integrate transformer architectures (e.g., BERT for network traffic logs or Graph Neural Networks (GNNs) for flow analysis) to enhance detection precision and evolve to changing VPN traffic patterns. Increase and Broaden the Dataset – Add additional real-world VPN traffic data, such as traffic from various VPN providers, protocols, and encryption algorithms, to accommodate growing cybersecurity threats. Deploy the Model at Scale– Move the model to a cloud or edge AI platform to enable real-time classification at scale across enterprise networks. Through the optimization of our machine learning method and the incorporation of new technologies, this system is designed to be a state-of-the-art solution for VPN classification, cybersecurity surveillance, and anomaly detection, providing strong privacy and security in today&#39; s networks.

References

[1] Ezra, P.J., Misra, S., Agrawal, A., Oluranti, J., Makelines, R. and Disamenities, R., 2022. Secured communication using virtual private network (VPN). Cyber security and digital forensics: proceedings of ICCSDF 2021, pp.309-319.

[2] M. F. Aziz, A. N. Khan, J. Shuja, I. A. Khan, F. G. Khan, and A. U. R. Khan, "A lightweight and compromise-resilient authentication scheme for IoTs," Trans. Emerg. Telecom Mun. Technol., vol. 33, no. 3, p. e3813, 2022. DOI: 10.1002/ett.3813.

[3] Kaur, D.C., 2022. The vital role of VPN in making secure connection over internet world. International Journal of Recent Technology and Engineering (IJRTE) ISSN, pp.2277-3878.

[4] Rashid, A. Masood, and A. U. R. Khan, "RC-AAM: Blockchain-enabled decentralized role-centric authentication and access management for distributed organizations," Cluster Compute., vol. 24, no. 4, pp. 3551–3571, Dec. 2021. DOI: 10.1007/s10586-021-03517-6.

[5] Wang, C., Yin, J., Li, Z., Xu, H., Zhang, Z. and Liu, Q., 2024, May. Identifying VPN Servers through Graph-Represented Behaviors. In Proceedings of the ACM Web Conference 2024 (pp. 1790-1799)..

[6] Brighten, D., Sisto, R. and Valenza, F., 2024. Automating VPN configuration in computer networks. IEEE Transactions on Dependable and Secure Computing.

[7] Maphumulo, A., Vermeulen, L., Pose, I. and Gasser, O., 2023, March. Characterizing the VPN Ecosystem in the Wild. In International Conference on Passive and Active Network Measurement (pp. 18-45). Cham: Springer Nature Switzerland.

[8] Setiawan, F., Chaniago, F.S. and Wibowo, A., 2024. Implements SSL VPN (Secure Socket Layer Virtual Private Network) Pada Badan Bank Tanah. Syntax Idea, 6(6), pp.2505-2516.

[9] Sharma, G., 2021. Secure remote access IPsec virtual private network to university network system. Journal of Computer Science Research, 3(1), pp.16-27.

[10] Chatterjee, P., Bose, R., Banerjee, S. and Roy, S., 2022. Secured Remote Access of Cloud-Based Learning Management System (LMS) Using VPN. In Pattern Recognition and Data Analysis with Applications (pp. 111-126). Singapore: Springer Nature Singapore.