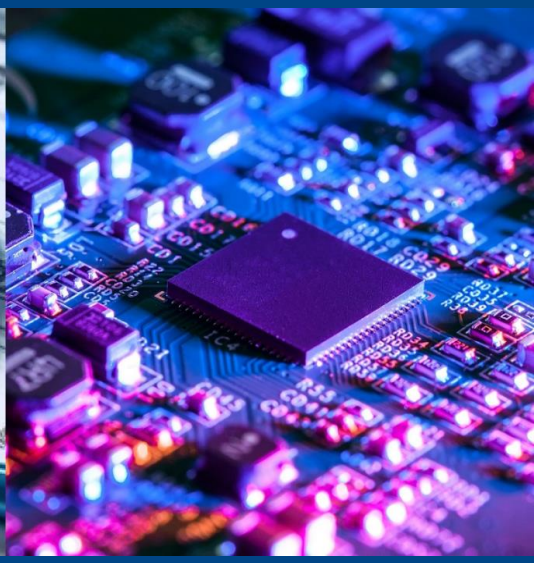
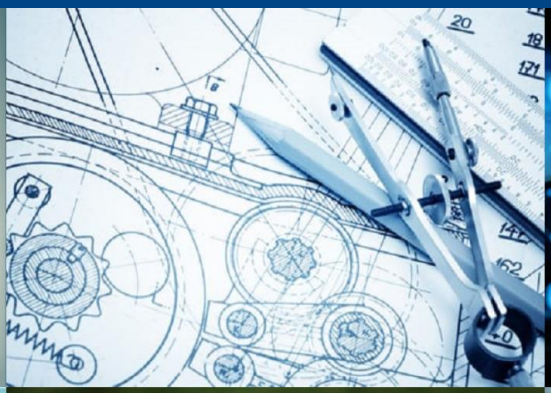
ISSN: 2582-7219

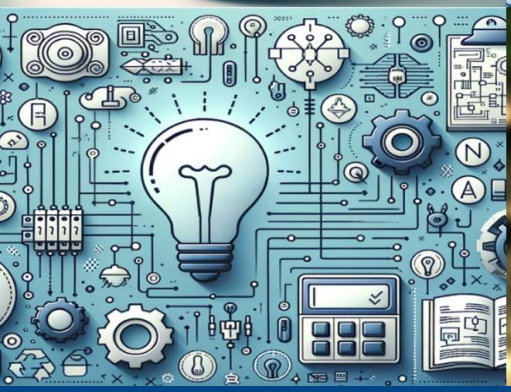




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 8, Issue 3, March 2025**

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803281

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Detection of Virtual Private Network Traffic using Machine Learning

Kavin NG, Dr.K. Devika Rani Dhivya

Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

ABSTRACT: Nowadays, VPNs are widely used to maintain privacy and bypass network restrictions, making it difficult to detect unauthorized users. Existing techniques face challenges in identifying VPN traffic, especially when users employ identity-masking tools such as anonymizing proxies and Virtual Private Networks (VPNs).
This study presents computational models to overcome these limitations in VPN detection. A Multi-Layered Perceptron (MLP) neural network was developed and trained using flow statistics extracted from the Transmission Control Protocol (TCP) headers of captured network packets. Validation testing showed that the proposed models effectively classify network traffic in a binary manner as direct (originating from a user's device) or indirect (masked through VPNs) with high accuracy.

KEYWORDS: VPN Classification, OpenVPN, Stunnel, Neural Networks, Machine Learning, Network Traffic Analysis, Feature Selection, Deep Packet Inspection, Traffic Encryption, Cybersecurity, TCP Flow Analysis, Wireshark, Weka, NetMate, Cross-Validation.

I. INTRODUCTION

Virtual Private Networks (VPNs) are widely used to conceal online activities, making them a tool for both legitimate and malicious purposes. Originally designed for secure remote access to enterprise networks, VPNs can also enable cybercriminals to hide their location while infiltrating systems. Notable incidents, such as the 2014 Sony Pictures hack and the LinkedIn data breach, highlight the potential misuse of anonymity tools, though the role of VPNs in these cases remains uncertain. Anonymity technologies, including mix networks, route data through multiple nodes to obscure the origin of communication. Low-latency systems like Tor, HTTP/SOCKS proxies, and VPNs are categorized into multi-hop and single-hop anonymous communication models. Open proxies, available to any user online, are commonly used for anonymous browsing, while VPNs establish encrypted tunnels between users and servers, enhancing security. To counter malicious network threats, IP blocking is a common technique. However, users can bypass restrictions by switching proxy or VPN providers. Access Control Lists (ACLs) and Deep Packet Inspection (DPI) further enhance security by filtering network traffic based on predefined policies. DPI, in particular, analyzes data packets to detect anomalies but is often circumvented by encrypted VPN traffic. Traditional methods of classifying internet applications, such as TCP and UDP port-based identification, have become less effective due to evolving application designs. Machine learning approaches are now being explored to improve detection. Packet headers and flow-based features serve as training data for classification algorithms, enabling more accurate identification of anonymous traffic. This research proposes a Multi-layered Perceptron Neural Network to detect and classify VPN traffic, enhancing network security by identifying potential threats in real time while minimizing false positives and negatives. While VPNs serve legitimate purposes, detecting their misuse can help mitigate cyber threats.[3]

Figure 1: Virtual Private Network (VPN)

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803281

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. VIRTUAL PRIVATE NETWORKS (VPNS)

A VPN enables private network access over public networks by tunneling traffic between remote and gateway machines. However, intercepted packets remain vulnerable unless encrypted and authenticated. Different VPN protocols offer varying levels of security.[1]

2.1 PPTP

The Point-to-Point Tunneling Protocol (PPTP) tunnels PPP connections through an IP network. While Microsoft's Windows NT implementation supports authentication and encryption negotiation, it relies on existing PPP extensions. Authentication methods include PAP, CHAP, MS-CHAPv1/v2, and EAP. PAP is highly vulnerable due to unencrypted credential transmission, and CHAP-based protocols have faced security scrutiny. Due to these weaknesses, PPTP is rarely used today.[5]

2.2 L2TP

The Layer 2 Tunneling Protocol (L2TP) enhances PPP by integrating PPTP and L2F features. It encapsulates PPP connections in UDP packets but lacks built-in encryption or authentication. Initial PPP methods were vulnerable to denial-of-service (DoS) attacks, later addressed in L2TPv3, which introduced optional authentication and integrity checks. L2TP is often paired with IPsec for stronger security.[5]

2.3 IPsec

IPsec authenticates VPN endpoints and negotiates encryption keys. It operates in two modes:

- **Transport Mode:** Adds an IPsec header to the original IP packet for authentication and integrity verification.
- **Tunnel Mode:** Encapsulates the entire IP packet inside a new one, ensuring confidentiality.

Security associations define encryption modes, algorithms, and keys, managed
through the Internet Key Exchange (IKE) protocol.[5]

2.4 OpenVPN

OpenVPN is widely regarded for its enterprise-grade security and cross-platform support. It ensures packet integrity using HMAC with SHA1 and offers two authentication modes:

- **Static Key Mode:** Uses a pre-shared key for authentication and encryption.
- **SSL/TLS Mode (Preferred):** Requires both hosts to authenticate via certificates before exchanging encryption keys. Keys are dynamically generated using OpenSSL's RAND_bytes function or TLS pseudorandom functions.

OpenVPN encrypts packets with the Blowfish cipher and tunnels data over UDP, maintaining a reliable transport layer for SSL/TLS sessions.[6]

III. VPN CLASSIFICATION

A dataset comprising TCP packets was generated using Wireshark, a packet analysis tool, to capture network traffic from an OpenVPN connection. The same machine learning techniques were applied using Azure's ML tools to analyze the dataset. The initial results revealed that the network exhibited overfitting, achieving a 100% classification accuracy for both VPN and non-VPN traffic. However, when subjected to external validation tests, the model performed poorly, effectively classifying all samples as VPN traffic.[1] This indicated that the model was merely guessing rather than learning meaningful distinctions between VPN and non-VPN traffic.

To mitigate this issue, an alternative dataset was proposed, consisting of TCP flow records and statistics. Flow statistics offer a more generalized view of network communications by tracking details such as addresses, ports, and byte/packet counts. This form of data becomes particularly useful in cases where encryption is used, as seen in VPN traffic. Similar to the first dataset, Wireshark was employed to capture packets for this refined dataset.[7]

The data collection process was executed using an Ubuntu 16.04 virtual machine hosted on a Windows 10 system. The network interface used for the experiment was a virtualized Intel PRO gigabit Ethernet card. Linux was selected due to its superior control over networking components, allowing automation of network connections and interface management. This capability proved beneficial when capturing VPN packets, as VPN connections generally begin with

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

a standard TCP "hello" sequence followed by key exchange. Once established, the connection persists until terminated by the user, forming a continuous TCP link between the user's system and the VPN server.[7]

3.1 OPEN VPN WITH STUNNEL

Stunnel is an open-source, multi-platform application that enables SSL/TLS encryption for clients and servers lacking native SSL/TLS support. While OpenVPN itself supports SSL/TLS, Deep Packet Inspection (DPI) techniques can detect OpenVPN traffic despite its encryption. Stunnel helps in circumventing this by presenting OpenVPN traffic as standard SSL web traffic over port 443.[11]

A key research question arose: could the same neural network model used for OpenVPN traffic classification be trained to recognize OpenVPN traffic encapsulated within Stunnel? To integrate Stunnel, both the OpenVPN server and client required installation and configuration of the application. This involved installing the stunnel4 package on Linux, generating and sharing OpenSSL certificates, modifying Stunnel configuration files, and configuring firewalls on both the server and client to facilitate Stunnel traffic transmission.[13]

3.2 DATASET COLLECTION

As with the previous experiment, a dataset capturing network traffic from Stunnel-based OpenVPN connections and regular non-VPN traffic was required for neural network training. Since an OpenVPN server was already established on AWS from prior experiments, setting up Stunnel was relatively straightforward. The Streisand VPN package, which includes built-in Stunnel support for OpenVPN, required only minor configuration modifications to be operational.
Once the VPN was set up and connections stabilized, network traffic capture commenced. Wireshark was again employed for packet collection, with the VPN being set to disconnect and reconnect at regular 10-minute intervals. An automated browsing script generated network activity by visiting a predetermined selection of websites. The captured packets were then processed using NetMate, a TCP flow analysis tool, to extract flow statistics. The final dataset contained a total of 3,952 samples: 1,931 representing Stunnel OpenVPN traffic and 2,021 representing non-VPN traffic.[13] The dataset was subsequently loaded into Weka for analysis.

3.3 FEATURE SELECTION

Feature selection was applied to the dataset to eliminate redundant attributes and retain only the most relevant ones. The Correlation Attribute Evaluation model in Weka was utilized, operating under a threshold of 0.5. The selected features differed significantly from those in the original VPN dataset. Some attributes, such as duration, reappeared but with altered correlation coefficients, suggesting that Stunnel introduces distinct modifications to OpenVPN traffic patterns.[12]

Attribute Name	Correlation Coefficient
min_fpktl	0.992
duration	0.937
max_fpktl	0.913
max_idle	0.780
max_biat	0.763
std_idle	0.719
max_fiat	0.673
mean_idle	0.575
min_idle	0.562
mean_fpktl	0.561
mean_active	0.512
max_active	0.511
std_fpktl	0.506

Table 1: displays the correlation coefficients for selected features:

After feature selection, the dataset was resampled into training, testing, and validation subsets. The training set comprised 3,160 samples, the testing set contained 633 samples, and the validation set included 127 samples.[13]

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803281

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3.4 NEURAL NETWORK SETUP

The neural network model from the previous experiment was reused to evaluate its effectiveness in classifying Stunnel-based OpenVPN traffic. The architecture consisted of a fully connected network with a hidden layer, where the number of nodes was determined by summing the number of attributes and classes and dividing by two. Given 13 attributes and 2 classes, the resulting computation yielded 7 hidden nodes.

The dataset was trained and tested using three different validation methods:

- **80/20 Split Validation** – The dataset was split into 80% training and 20% testing data.
- **10-Fold Cross-Validation** – The dataset was divided into 10 subsets, with each serving as validation data in successive iterations.
- **Leave-One-Out Cross-Validation (LOOCV)** – Each sample was used once for validation while the rest formed the training set.

Initial results suggested potential overfitting, prompting adjustments to the learning rate and momentum, reducing both from 0.1 to 0.01 to enhance generalization.[16]

3.5 Results and Performance Evaluation

Metric	80/20 Split	10-Fold Cross Validation	LOOCV
Accuracy	98.42%	97.89%	97.82%
True Positive Rate	0.968	0.969	0.968
False Positive Rate	0.000	0.012	0.012
Precision	1.000	0.987	0.987
Recall	0.968	0.969	0.968
F-Measure	0.984	0.978	0.978

Table 2: Represent the results of each validation method.

The confusion matrices for each test method showed high classification accuracy with minimal false positives. The 80/20 split validation method demonstrated the best accuracy at 98.42%, followed closely by the cross-validation methods.[16]

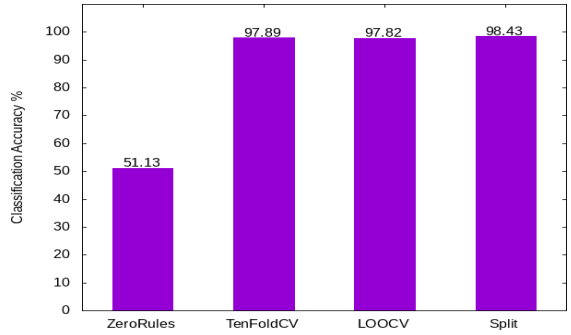


Figure 2: Graph comparing accuracies of different validation techniques.

The confusion matrices for each test method showed high classification accuracy with minimal false positives. The 80/20 split validation method demonstrated the best accuracy at 98.42%, followed closely by the cross-validation methods.[12]

The findings indicate that while neural networks can effectively classify OpenVPN traffic, additional techniques such as Stunnel may slightly alter traffic characteristics, necessitating refined classification models. Future research could explore alternative deep learning architectures or feature engineering techniques to further enhance classification robustness.[12]

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. CONCLUSION

The aim was to investigate methods that would aid in the detection of VPN technologies that are being used to hide an attacker's identity. While VPNs have legitimate uses, such as connecting to a business network from a remote location, they are still abused by criminals who use them to commit crimes whilst remaining undetected and unidentified. Without a method to identify when a VPN is connecting to a web facing server, businesses could be vulnerable to having their network breached and having data stolen whilst being hindered in their ability to confidently say who stole it. This can be particularly detrimental to websites who deal with customer details and financial records. There are methods available for inspecting network traffic at the point of ingress and egress. [2] An example of one of these methods is Deep Packet Inspection (DPI). It is closely related to another method called Shallow Packet Inspection (SPI), however SPI only has the ability to inspect the headers of network packets that are used to transport the packets to their destination. DPI goes a step further and inspects those headers and the actual content of the packet, which in the case of a HTTP packet could be a request for data from a website. A counter to DPI is the use of end-to-end encryption on the content of packets in order to hide those contents from prying eyes. This is done innocently enough with the goal being to stop potential man in the middle attacks from stealing sensitive data such as usernames and passwords or financial details as they are being transmitted. However, proxy and VPN technologies also have the ability to use encryption technologies with the use of IPSec and SSL/TLS. This increases the need for a method to identify these types of network traffic. Machine learning techniques are one way in which to accomplish this.[5] The experiments conducted to classify OpenVPN usage found that the Neural Network was able to correctly identify the VPN traffic with an overall accuracy of 93.71%. The further work done to classify Stunnel OpenVPN usage found that the Neural Network was able to correctly identify VPN traffic with an overall accuracy of 97.82% accuracy when using 10-fold cross validation. This final experiment also provided an observation of 3 different validation techniques and the different accuracy results obtained. Upon successful experiments conducted for the detection of Anonymising Proxy traffic, the focus was extended to include VPN traffic. The VPN technology OpenVPN was chosen as the focus for the experiments, which in turn found that the Neural Network was capable of classifying network traffic as either VPN traffic or as non-VPN traffic. [6] This led to a further set of experiments which attempted to classify a form of OpenVPN traffic that made use of Stunnel to provide encryption. These found that a Neural Network trained on the Stunnel OpenVPN data could classify network traffic as either VPN traffic or non-VPN traffic. [7] Again, the experiments were conducted in such as fashion as to eliminate bias where possible. This included keeping a portion of the captured dataset away from the training and tuning phases, so it could be used to simulate real world data that the model had never seen before.[10]

REFERENCES

[1] Alshammari, R., & Zincir-Heywood, A. N. (2011). "An investigation of patterns in encrypted traffic." 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 146-153.
[2] Bujlow, T., Carela-Español, V., Solé-Pareta, J., & Barlet-Ros, P. (2015). "A survey on web tracking: Mechanisms, implications, and defenses." Proceedings of the IEEE, 105(8), 1476-1510.
[3] Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., & Salamatian, K. (2006). "Traffic classification on the fly." ACM SIGCOMM Computer Communication Review, 36(2), 23-26.
[4] Coull, S. E., & Dyer, K. P. (2014). "Traffic analysis of encrypted communications: The state of the art." IEEE Communications Surveys & Tutorials, 18(1), 204-221.
[5] Anderson, B., Paul, S., & McGrew, D. (2016). "Deciphering malware's use of TLS (without decryption)." Journal of Computer Virology and Hacking Techniques, 14(3), 195-211.
[6] Dainotti, A., Pescape, A., & Claffy, K. (2012). "Issues and future directions in traffic classification." IEEE Network, 26(1), 35-40.
[7] Wright, C. V., Monrose, F., & Masson, G. M. (2006). "On inferring application protocol behaviors in encrypted network traffic." Journal of Machine Learning Research, 7(Dec), 2745-2769.
[8] Zhang, K., Zheng, H., Wang, S., Luo, P., & Wang, J. (2019). "Detecting VPN traffic using machine learning techniques." IEEE Access, 7, 102857-102867.
[9] Jaber, R., Hassan, R., Tahir, M., & Naeem, U. (2020). "Deep learning-based VPN traffic classification: An experimental study." IEEE Access, 8, 22919-22931.
[10] Rezaei, S., & Liu, X. (2019). "Deep learning for encrypted traffic classification: An overview." IEEE Communications Magazine, 57(5), 76-81.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[11] Siracusano, G., Gallo, M., Bifulco, R., & Di Pietro, R. (2021). "Feature selection for encrypted traffic classification: An experimental comparison." Journal of Network and Computer Applications, 188, 103091.
[12] Velan, P., Čermák, M., Čeleda, P., & Drašar, M. (2015). "A survey of methods for encrypted traffic classification and analysis." International Journal of Network Management, 25(5), 355-374.
[13] Bernaille, L., Teixeira, R., & Salamatian, K. (2006). "Early application identification." Proceedings of the 2006 ACM CoNEXT Conference, pp. 1-12.
[14] Auld, T., Moore, A. W., & Gull, S. F. (2007). "Bayesian neural networks for internet traffic classification." IEEE Transactions on Neural Networks, 18(1), 223-239.
[15] Zhang, S., Chen, Z., Wang, W., Pei, Y., & Liu, Z. (2019). "A survey on encrypted traffic classification." IEEE Communications Surveys & Tutorials, 21(3), 2450-2473.
[16] Wang, W., Zhu, M., Wang, J., Zeng, X., & Zhu, Z. (2018). "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection." IEEE Access, 6, 1792-1806.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com