AI-Enhanced VPN Security Framework: Integrating Open-Source Threat Intelligence and Machine Learning to Secure Digital Networks

Mohamad Hasan and Dr Tania Malik
School of Informatics and Cybersecurity, Technological University Dublin, Ireland

b00158228@mytudublin.ie
Tania.Malik@tudublin.ie

Abstract: In today's digital age, ensuring network privacy and integrity is of utmost importance. To address this, our work proposed an advanced VPN security framework that integrates open-source threat intelligence and machine learning (ML) to enhance cyber defences. By combining Wazuh for threat detection and analysis, and pfsense for firewall capabilities, with state-of-the-art ML algorithms, we present a robust VPN security solution to the challenges presented by the evolving landscape of cyber threats, representing a significant advancement in securing digital networks. This framework is strengthened by the integration of four ML algorithms— Gradient Boosted Trees (GBT), Random Forest (RF), K-Nearest Neighbors (KNN), and Dense Deep Learning (DDL)— chosen for their classification efficacy and their ability to process complex security data, thereby improving the efficiency and accuracy of threat detection. Results indicated significant improvements in threat detection accuracy following the integration of ML algorithms. The Random Forest (RF) algorithm, in particular, stood out for its exceptional accuracy and ability to handle various threat scenarios, showcasing its efficacy in identifying sophisticated cyber threats through network traffic pattern analysis. Further performance benchmarking confirmed the feasibility of deploying the advanced VPN security framework, demonstrating minimal impact on network latency and throughput.

Keywords: VPN Security, ML in Cybersecurity, Deep Learning, Encrypted Traffic, VPN Framework

1. Introduction

Network privacy and integrity has become the utmost priority in today's digital world. The internet not only facilitates global communication and access to worldwide events but is increasingly supplanting traditional business practices. With the rise of remote working, there has been a significant shift away from the secure environments of corporate offices, leading individuals and businesses to frequently rely on Virtual Private Networks (VPNs) to access geographically restricted resources. Ensuring the data security and the reliability of networks in such settings is paramount. VPNs create secure tunnels between remote users and networks, protecting data traffic from external threats. However, the security of these tunnels depends on the robustness of the tunneling protocols and the integrity of the network devices. Despite their popularity, VPNs are not immune to vulnerabilities. Current VPN setups often face security challenges that can compromise sensitive data and threaten network integrity (Iqbal, MA and Riadi, I. 2019). These vulnerabilities are intensified by the rapidly evolving and increasingly complex landscape of cyber threats.

Given the vulnerabilities identified in existing VPN configurations and the evolving landscape of cyber threats, there is a pressing need for a more advanced VPN security framework. To address this, we proposed an advanced VPN security framework that integrates open-source threat intelligence and machine learning (ML) to enhance cyber defense. The core objective of our research is to develop a robust VPN security framework that enhances the detection and mitigation of internal network attacks. These attacks typically become evident after remote PCs are connected to the network via a VPN, exposing the network to various security risks. This AI-enhanced VPN security framework aims to deliver a comprehensive solution that secures VPN connections as an integral part of the VPN service, independently of external network security devices. In this advanced framework, we integrate Wazuh for threat detection and analysis, and pfSense for firewall capabilities, along with cutting-edge ML algorithms to create a robust VPN security solution that addresses the challenges posed by the dynamically evolving cyber threat landscape, marking a notable advance in digital network security.

The framework is enhanced by incorporating four algorithms—Gradient Boosted Trees (GBT), Random Forest (RF), K-Nearest Neighbors (KNN), and Dense Deep Learning (DDL). These algorithms were specifically chosen for their high classification efficacy and their ability to process complex security data, significantly improving the efficiency and accuracy of threat detection. The major contribution of this research is the integration of machine learning with open-source threat intelligence and firewalls to create a robust, standalone solution that can be implemented within any network without the need for additional devices to monitor network traffic, all through an integrated VPN solution. To the best of our knowledge, this is the first research effort to apply ML specifically

to VPN security, as most previous research has focused only on analyzing encrypted traffic and identifying attack patterns within it. In our study, we also evaluate which model performs best for this specific issue. Our experimental results demonstrate significant improvements in threat detection accuracy following the integration of these ML algorithms. Additionally, our performance benchmarking has confirmed the feasibility of implementing this advanced VPN security framework, demonstrating minimal impact on network latency and throughput.

2. Literature Review

Recent developments in VPN security have increasingly incorporated machine learning techniques to enhance encryption and threat detection capabilities. In this literature review we cover various studies that have significantly contributed to the application of ML in the field of VPN security, particularly focusing on the classification of encrypted traffic and the improvement of security protocols.

Significant efforts have been explored in utilizing traditional ML algorithms to better understand and secure VPN traffic. For instance, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes have been widely used for their ability to classify and predict outcomes based on historical data, which is crucial for identifying anomalous patterns that may indicate security breaches (Zhou, Y. et al. 2023), (Bagui, S. et al.2017). Furthermore, the k-Nearest Neighbors (kNN) algorithm has been applied for its simplicity and effectiveness in classification tasks by comparing new data points with known data points (Bagui, S. et al. 2017). Additionally, ensemble methods such as Random Forest and Gradient Boosted Trees have received attention for their robustness and accuracy in handling complex datasets with multiple input variables. These methods combine multiple decision trees to improve the predictive performance and have proven to be particularly effective in enhancing the detection capabilities within VPN security frameworks (Wang, Z. et al. 2022), (Muliukha, V.A. et al. 2020). These models were compared to determine which could most effectively classify the encrypted network traffic with high accuracy and minimal overfitting. Among the models tested, ensemble methods, particularly Random Forest and Gradient Boosting Trees, were highlighted for their superior performance in terms of accuracy and the ability to handle overfitting (Bagui, S. et al. 2017). Muliukha et al. (2020) discussed the application of machine learning algorithms such as Random Forest and Naive Bayesian for the classification of encrypted network traffic, specifically focusing on SSL sessions and VPN connections. Similarly, Wang et al. (2022) provided a comprehensive analysis of using machine learning to detect encrypted malicious traffic, offering insights into different datasets and the efficacy of various ML algorithms in this context. The importance of robust datasets for training and testing ML models in traffic classification was emphasized by Uğurlu et al. (2021), who developed a new classification method that leverages extreme gradient boosting (XGBoost) and other ML techniques. They reported a high success rate in classifying encrypted traffic, which underscores the potential of ML in enhancing VPN security.

Encrypted traffic analysis has become more sophisticated with the application of deep learning models. Works by Zhou et al. (2023) introduce one-dimensional convolutional neural networks for efficient and accurate traffic classification. Research by Naas and Fesl (2023) introduced a novel dataset for VPN traffic analysis, which could significantly aid in the development of ML models tailored for VPN security. This dataset includes various VPN protocols and is designed to help researchers improve network Quality of Service (QoS) and security features. Several studies have focused on the detection of malicious activities through encrypted traffic. For instance, Wang et al. (2022) and Zhao et al. (2013) have explored machine learning-based approaches for identifying malicious entities within encrypted traffic, which is crucial for preventing security breaches in VPNs. The integration of ML algorithms into network security systems has been proven to significantly reduce false positives, thereby enhancing the operational efficiency of cybersecurity measures. This is particularly relevant in the context of VPN security, where distinguishing between benign and malicious traffic quickly is crucial. However, existing research predominantly focuses on the analysis of encrypted traffic, seeking patterns within this data—a process that is inherently complex. Once a user connects through the VPN gateway, the traffic is decrypted on the internal side, where analyzing it requires substantially lower processing capabilities and allows for much easier implementation with a focus on effective monitoring of internal network traffic.

Our research introduces an integrated framework dedicated to VPN security, based on machine learning and open-source threat intelligence. This framework is designed to monitor any suspicious behaviour of connected users and provide feedback if their actions are deemed malicious. The selection of machine learning algorithms in this study is based on (Bagui, S. et al. 2017) work that compared various ML algorithms used in network traffic analysis, focusing on their performance, accuracy, and precision. This approach facilitates a more efficient and straightforward implementation while maintaining high levels of security and threat detection.

3. Framework Overview

In this section, we discussed in detail the advanced VPN security framework's architecture and its functionality. The framework integrates multiple components—Wazuh, pfSense, and several machine learning (ML) algorithms—to create a robust solution for mitigating the evolving landscape of cyber threats.

3.1 Architectural Design

A Graphic representation of the architecture shown in Figure 1 illustrates how data flows between the threat detection modules, firewall, and ML processing units. This design ensures that each component can operate efficiently while maintaining high security standards across the network. At first, we discuss the specifics of each component within the framework, detailing their roles and contributions to the overall functionality and security capabilities of the system. Each component is crucial for ensuring the robustness and efficiency of our advanced VPN security framework. After that, we describe the implementation detail of the framework and data flow and processing interactions among the components of the framework.

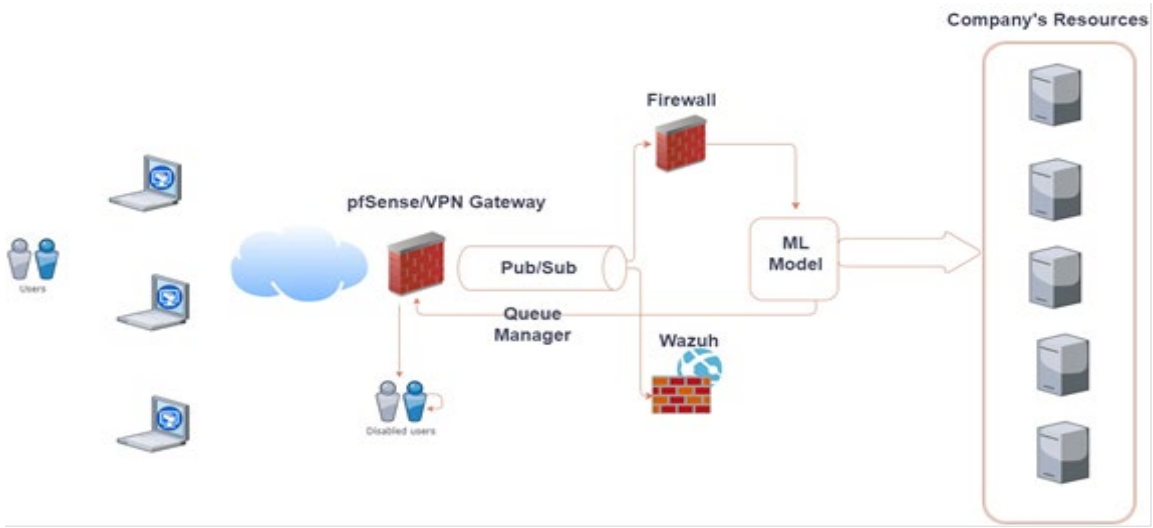

Figure 1: AI-Enhanced VPN Security Framework

3.1.1 Wazuh

Within our VPN security framework, Wazuh acts as a critical component. Wazuh (Asswad A2022) is an open-source security monitoring tool that provide comprehensive protection capabilities and built on the Open-Source Host-based Intrusion Detection System Security (OSSEC) framework, providing features like intrusion detection, log analysis, and file integrity monitoring. Wazuh excels at aggregating logs from various points within the network, including the pfSense/VPN gateway, offering a unified view of the system's activities. Its strength lies in the real-time analysis of collected data, utilizing advanced rule-based threat detection algorithms that can recognize signs of malicious activity, such as intrusion attempts or malware presence. By correlating and analyzing the patterns within the traffic data, Wazuh contributes significantly to the proactive aspect of the framework, enabling early detection of potential threats before they can escalate into breaches. This integration ensures that any abnormal or suspicious behaviours are flagged promptly, strengthening the VPN network's defense against cyber threats and aligning with the framework's goal of maintaining a secure and resilient digital environment for any organization.

3.1.2 pfSense

Another component of the VPN security framework is pfSense which acts as the VPN gateway, and it is therefore the first perimeter of defense. pfSense (Patel, at al 2017) is an open-source and FreeBSD-based firewall and router platform with number of features, among which is the capability for VPN provision. Inside, pfSense is the VPN gateway—managing secure access to the company network and the encrypted/decrypted traffic. This is highly valuable functionality for the framework, which is responsible for steering the flow of data through

security checks to the machine-learning models. Also, at the second stage it acts as a firewall where we can use it's Suricata threat detection plugin (Meena, at al 2020) as another layer of protection.

3.1.3 ML Algorithms

This section provides a detailed description of our chosen machine learning algorithms and the rationale behind choosing them.

- Random Forest (RF) - The Random Forest algorithm (Zhou, Y. et al. 2023, Bagui, S. et al. 2017) is an ensemble learning method used for classification tasks, which constructs multiple decision trees during training and outputs the class that is the mode of the classes predicted by individual trees. It aims to improve prediction accuracy and control over-fitting by averaging the results of several trees, each created on randomly selected subsets of the data. Adjusting the number of trees in the forest is key to finding the right bias-variance balance, with more trees leading to better performance. Although less directly interpretable compared to some other methods, Random Forest requires minimal parameter tuning and relies on an understanding of the data to yield effective outcomes, making it a practical and robust choice for predictive analysis.
- Gradient Boosted Trees (GBT) - Gradient Boosting Trees (GBT) (Bagui, S. et al. 2017) is a powerful ensemble learning technique used for both regression and classification problems. Like Random Forest, GBT builds multiple decision trees, but in a sequential manner where each subsequent tree aims to correct the errors of the previous ones. This method combines weak predictive models to create a strong model in a stage-wise fashion, using an optimization algorithm that minimizes a loss function. GBT is particularly effective because it focuses on areas where the previous trees performed poorly, iteratively improving the model's accuracy. The model uses gradient descent to minimize errors, which involves adjusting tree parameters to better predict the outliers and misclassified data points from earlier iterations. The result is a highly accurate prediction model that can handle complex non-linear relationships within the data.

- K-Nearest Neighbors (KNN) - K-Nearest Neighbors (KNN) is a straightforward and widely used classification algorithm (Zhou, Y. et al. 2023, Bagui, S. et al. 2017, Lichy, A. et al. 2022) that predicts the label of a data point by looking at the 'k' closest labeled data points and choosing the most common label among them. This non-parametric method operates by calculating the distance (often Euclidean) between points and aggregating the categories of nearest neighbour data points. KNN is unique in that it doesn't build a model in advance but performs the classification directly at the time of prediction, making it very intuitive and easy to implement. Despite its simplicity, KNN can be highly effective, particularly in cases where the decision boundary is irregular. To enhance performance, the choice of 'k' and the distance metric can be adjusted based on the specific characteristics of the data.

- Dense Deep Learning (DDL) - Dense Deep Learning (DDL) (de Brito Guimarães, L.C. 2023) involves the use of densely connected neural networks, a type of deep learning architecture where each neuron in a layer is connected to every neuron in the next layer, maximizing the potential for learning complex patterns in data. These models are particularly known for their ability to extract features at multiple levels of abstraction, allowing them to capture intricate dependencies and relationships within the data. The strength of DDL lies in its layered structure, which can include hundreds or even thousands of layers, each contributing to a more refined analysis of the input features. This makes Dense Deep Learning highly effective for tasks involving large volumes of data with many variables, such as image and speech recognition, and complex decision-making scenarios in real-time systems.

3.2 Configuration and Implementation

After describing overall architecture and individual components of VPN security framework, we will now discuss the specific configuration steps and detailed implementation of these components within the framework.

Step 1: Configuration and Setup of PfSense as a VPN Gateway

Setting up an OpenVPN server in pfSense involves several key steps to ensure secure and efficient operation. All the components of the framework are setup on virtual box as VM environment and was interfaced together as different components. First, a Certificate Authority (CA) is generated to authenticate the server and client credentials. Next, server and client certificates are created within pfSense's certificate manager. The OpenVPN server configuration is then established, specifying parameters like server mode, cryptographic settings, and

tunnel settings. After configuring the server, necessary firewall rules are defined to allow traffic through the specified ports. Lastly, the OpenVPN Client Export Utility is used to generate and distribute client configuration files. This streamlined process ensures that users can securely access their network remotely through the VPN.

Step 2: Wazuh Deployment

Wazuh was implemented as a Docker container on a Linux virtual machine within the VirtualBox environment. Docker was chosen for its ability to simplify the deployment process. The Wazuh container was linked to the network where VPN traffic flows to monitor logs in real-time. Configuration files were tailored to define the specific security rules and alerts relevant to the traffic patterns of the organization. The Wazuh manager was set up to aggregate logs, analyse data, and flag potential security incidents, acting as an SIEM (Security Information and Event Management) system.

Step 3: Setting Up and Configuring a Second pfSense Instance for Enhanced Security

A second pfSense instance was configured to act as a dedicated firewall within the VirtualBox environment. This instance was responsible for filtering incoming and outgoing VPN traffic after the VPN gateway in order to minimize CPU resources usage within the firewall. Network interfaces were carefully configured to establish distinct zones for internal which is connected to a virtual server and external traffic which is coming from the vpn gateway. Firewall rules were set to allow legitimate traffic from a specific subnet dedicated to VPN users and block malicious or unauthorized access. This setup was crucial for creating a strong security layer that would work in parallel with the VPN gateway.

Step 4: Integration of Argus for Network Monitoring

Argus (Audit Record Generation and Utilization System) has been integrated with the audit network tool very methodically, capturing details of network traffic from the internal interface of the firewall. Argus provides a network-wide visibility tool with a fine perspective since it offers a fine-grain perspective on network activity by generating complete flow metrics data. Argus captures the data at the packet level, providing visibility into the details of what type, structure, and behavior of traffic is going by the firewall. It should monitor and report the state of network transactions, systematically logged for analysis.

Step 5: Implementing and Deploying Machine Learning Model

The machine learning component is developed using Python, with scripts written for training and testing the four algorithms: GBT, RF, KNN, and DDL which are described in above section. The Python environment was set up with necessary libraries, we used scikit-learn for algorithm implementation and NumPy for data manipulation. The scripts were executed within a Linux VM hosted on VirtualBox, accessing pre-processed traffic data to train the models. Once trained, the models were tested for performance and accuracy in classifying traffic as benign or malicious. The captured packet data is a vital resource for the machine learning (ML) algorithms at play within the framework. The ML algorithm ingests rich, granular flow data from Argus, including metrics on duration, packet counts, byte counts, and transaction sequences as the traffic data streams in. This data is then used by the algorithm for understanding normal behavior within patterns of the network traffic so that it would then be possible to detect changes, which would in turn show potential security incidents or malicious activity.

3.3 Data Flow and Processing

The VPN security framework begins with a thorough analysis of all potential threats. This involves assessing the threat agents that might be interested in accessing the data protected by the VPN, which is essential for developing an effective threat model. Threat modeling is a complex process that encompasses a broad spectrum of potential threats and vulnerabilities. The general approach includes identifying a structured list of all conceivable threats, envisaging the worst-case scenarios, and devising prevention strategies. This methodology is valuable not only for setting security benchmarks but also for prioritizing threats and vulnerabilities based on their potential impact. Threat modeling is a dynamic process; as new threats emerge; the VPN's design may need to be adjusted. Any modifications to the VPN's design and security policies should be evaluated to determine whether they reduce or exacerbate the risk of threats. Static analysis of the VPN's configuration data can reveal vulnerabilities, while dynamic analysis involves simulating potential attack scenarios and assessing the effectiveness of threat detection and prevention strategies. The outcomes of this analysis could lead to modifications in the VPN's security policy and adjustments in network and security configurations. An ongoing cycle of evaluation and feedback helps enhance the efficacy of the security policy, with mechanisms in place to

detect and address any violations or unintended configuration changes, ultimately improving the security policy's quality.

In our advanced VPN security framework, the VPN user connects to the WAN interface of the pfSense firewall and initiates a VPN connection. Once authenticated, all logs are sent to Wazuh for analysis as a syslog server, and the traffic is routed through a Linux server acting as a router. This is facilitated by configuring a third pfSense interface as a SPAN port, which mirrors all traffic from other interfaces. The Linux machine, connected to this virtual interface, runs an ML script on the traffic captured, segmenting it minute-by-minute into a CSV file stored in the "tmp" folder. If malicious activity is detected, the ML script triggers an alert. Meanwhile, another pfSense instance functions as a standard firewall, managing policies that allow VPN users' IP subnet to communicate with servers on the internal network, and preventing these servers from initiating traffic to the WAN interface to enhance protection of internal resources. Wazuh is configured to automatically send an email alert when a threat is detected as shown in Figure 2. This real-time alert system contrasts with the machine learning section's minute-by-minute CSV analysis. For demonstration purposes, two internal servers were set up and connected to the internal interface to illustrate complete traffic flow. The efficacy of the system was tested using two methods: deploying Metasploit from Kali Linux to identify CVEs and utilizing a 7-day demo of Tenable, both of which are reliable tools for vulnerability testing. These tools help to confirm when the traffic is identified as attack traffic by any threat intelligence system.

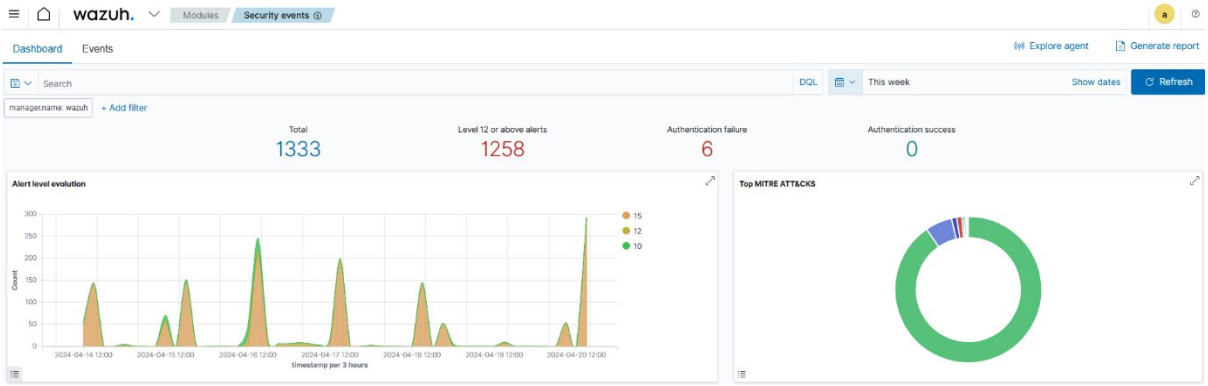

Figure 2: Wazuh Dashboard after simulating attacks

4. Experimental Results

In our study, we have designed two types of experiments: one to evaluate and compare the accuracy and performance of each chosen machine learning model, and another set of experiments to assess the system impact analysis of our VPN security framework. In this section, we will discuss the results of both experiments.

4.1 Performance and Accuracy Experiments

The first set of experiments was conducted to identify the most effective machine learning technique for detecting malicious activities in VPN traffic. Four ML algorithms were chosen for comparison—Gradient Boosted Trees (GBT), Random Forest (RF), K-Nearest Neighbors (KNN), and Dense Deep Learning (DDL). These were primarily selected for their classification efficacy and their ability to process complex security data, thereby enhancing the efficiency and accuracy of threat detection. The goal of these experiments is to evaluate the performance and accuracy of each selected ML model in identifying potential threats separately, to determine which algorithm—or combination of them—yields the best results. This comparison is crucial for optimally designing the security measures for the VPN network to ensure that the chosen model or models achieve the highest possible accuracy in classifying and mitigating cyber threats.

In our supervised learning environment, we were looking for a rich dataset that is big enough to include various types of known attacks such as DOS, Backdoors, Reconnaissance and Worms, also to be recorded over a decent period of time. Our digging came across the UNSW-NB15 Dataset (Moustafa, N. et al. 2015) which contains all of that and more, over a 175k training set and 82k testing set collecting in CSV files and also was captured using Argus so that adjusting the data to match the training data would be a smooth process. A quick check on the Dataset showed that there were some categorical values that needed to be dealt with as well as some feature

selections and feature removal to keep only the packets actual data relative features, this all was done using python script and tested to make sure that there will be no problem during either training or testing phases.

The expected impact on learning algorithm performance is important for security data, as we would like threat detection algorithms to not only learn the same pattern with less computation, but also to learn it with increased accuracy. An increased understanding of the relationship between specific optimization methods and algorithm performance could lead to the development of better optimization methods for specific algorithm types and the automatic selection of optimization methods for data sets of a specific type. That's why in our experiments we focused on optimising the features before using each algorithm experiment as well as testing in multiple environments (local and Google Colab cloud environment). The results were consistent across both platforms. We observed that some algorithms outperformed others in terms of accuracy and performance. We believe this performance can be attributed to the process of analyzing packets post-decryption through the internal interface of the firewall, which not only consumes fewer system resources but also enhances the efficiency of the algorithms. Using the DDL algorithm at the first time showed very low accuracy so we had to try several values for Epochs and Batch size during the training phase to get the best accuracy out of it, which occurred at Epoch=10 with batch size =8 and a split of 0.2.

Our findings indicate that in terms of performance, the Random Forest (RF) algorithm surpassed other models, with the Gradient Boosting Trees (GBT) algorithm close behind, performing only 1% lower. The K-Nearest Neighbors (KNN) algorithm ranked third, achieving a score above 80%, as depicted in Figure 3. Regarding accuracy, the RF algorithm again outperformed the rest, with GBT in second place, while KNN displayed the lowest accuracy, as illustrated in Figure 4.

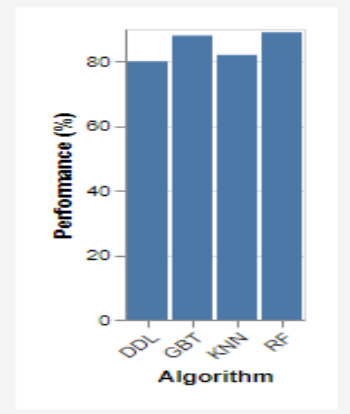

Figure 3: Comparison in terms of Performance

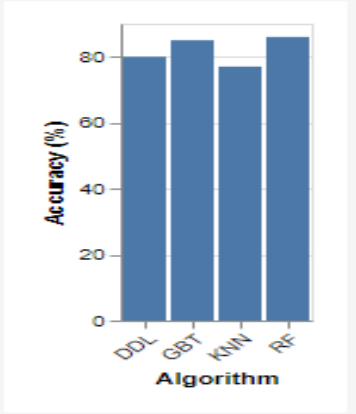

Figure 4: Comparison in terms of Accuracy

The comparison will not only to see how accurate each model is in the prediction but, also to check how each of the models has the capability to adapt and learn from the data. In this scenario, the Random Forest (RF) algorithm becomes one of the more attractive ways to extract great insight into network traffic, which will eventually improve its overall effectiveness with reference to the VPN security system.

4.2 System Impact Analysis

The second set of experiments evaluated the system's impact, first conducted within a VirtualBox environment on a Windows machine equipped with 32 GB of RAM, an Intel Core i7 CPU at 3.2 GHz, and SSD storage. While the overall performance was not considered slow, some delays were noted during the training phases of the KNN and DLL algorithms. However, no performance issues were observed during testing, with traffic type notifications displayed within 3 seconds. Despite this, we opted to transfer the machine learning workload to the Google Collab environment to assess how the models would perform on the cloud, utilizing a T4 GPU and 15 GB of RAM. The tests in this environment mirrored the performance and accuracy observed on the local machine. We theorize that to get the actual system performance impact; it should be deployed in a production environment with significant user traffic. Our findings suggest that the system can smoothly process up to 100 users' traffic, recorded into a CSV file, on an Intel Core i7 CPU without noticeable performance degradation. CPU and memory usage comparisons were conducted in the Google Collab environment using a Tesla T4 GPU, which features a clock speed of 1.59 GHz, 40 cores, and 16 GB of RAM. This GPU, though architecturally similar to a

CPU, offers significantly faster processing speeds. GPU and memory utilization of each algorithm is illustrated in Figure 5 and 6 respectively.

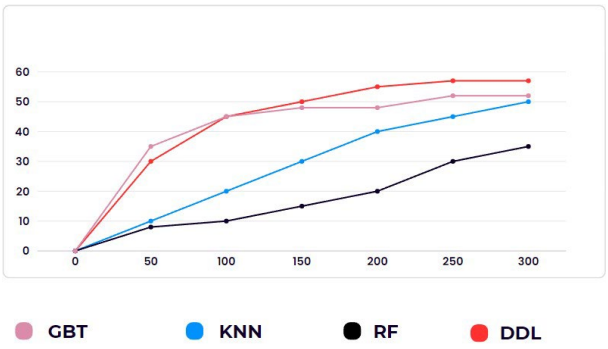

Figure 5: GPU Utilization for each algorithm

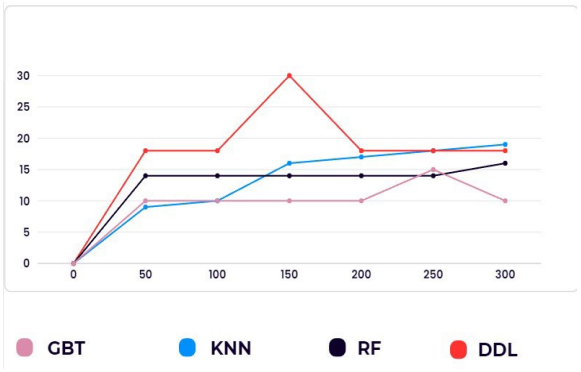

Figure 6: Memory Utilization for each algorithm

The machine learning algorithms GBT, KNN, RF, and DDL exhibit different patterns of GPU and memory utilization due to their varying operational complexities. DDL tends to steadily use more GPU over time because of its complex network layers, whereas RF does not utilize the GPU as much, due to its reliance on multiple decision trees. GBT also experiences high GPU usage, reflecting the computational demands of iterative error correction. KNN maintains moderate GPU and memory usage as it directly operates on the data during prediction. Figure 6 shows trends in memory utilization, suggesting that DDL requires more memory, likely for handling extensive data through its layers, while RF and GBT use memory more consistently—RF for storing multiple trees and GBT for managing tree computations. These trends highlight the inherent balance between algorithm complexity and resource efficiency, which is crucial for selecting algorithms in practice and also demonstrate that accuracy is not always correlated with algorithm complexity.

5. Conclusion

Machine learning (ML) algorithms are reshaping the way complex security data is evaluated. Even though ML methods are not a new concept in computer network defense, they have only recently been applied to the problem of intrusion detection and prevention. Machine learning algorithms have the potential to not only help secure VPN traffic more effectively but also reduce the total cost of ownership of security for an organization. Our work proposed an advanced VPN security framework that integrates open-source threat intelligence and machine learning models to enhance cyber defences. By combining Wazuh for threat detection and analysis, and pfsense for firewall capabilities, with state-of-the-art ML algorithms, we present a robust VPN security solution to the challenges presented by the evolving landscape of cyber threats, representing a significant advancement in securing digital networks. Our experiments demonstrate that these algorithms are highly effective in detecting anomalous behavior within network traffic. Particularly, the Random Forest (RF) algorithm excelled in accuracy and versatility across various threat scenarios, proving its effectiveness in identifying sophisticated cyber threats through analysis of network traffic patterns. Further performance benchmarking confirmed the viability of deploying this advanced VPN security framework with minimal impact on system.

Our future work will focus on integrating the ML component with a web service to enable real-time ML feedback, moving away from recording packet captures with Argus to a fully automated system that provides immediate notifications. Additionally, we plan to integrate the output from the ML algorithms directly into the firewall management process using the pfSense API, allowing for automatic blocking or permitting actions based on whether a traffic is identified as malicious.

References

Iqbal, Muhammad & Riadi, Imam. (2019). Analysis of Security Virtual Private Network (VPN) Using OpenVPN. International Journal of Cyber-Security and Digital Forensics. 8. 58-65.

Zhou, Y., Shi, H., Zhao, Y., Ding, W., Han, J., Sun, H., Zhang, X., Tang, C., Zhang, W. (2023) "Identification of encrypted and malicious network traffic based on one-dimensional convolutional neural network", Journal of Cloud Computing, 12(53).

Wang, Z., Fok, K.W., Thing, V.L.L. (2022) "Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study", Computers & Security, 113, p.102542.

Bagui, S., Fang, X., Kalaimannan, E., Bagui, S., Sheehan, J. (2017) "Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features", Journal of Cyber Security Technology, 1, pp. 1-19.

Muliukha, V.A., Laboshin, L.U., Lukashin, A.A. (2020) "Analysis and Classification of Encrypted Network Traffic Using Machine Learning", IEEE.

Uğurlu, M., Doğru, I.A., Arslan, R.S. (2021) "A New Classification Method for Encrypted Internet Traffic Using Machine Learning", Turkish Journal of Electrical Engineering & Computer Science.

Naas, M., Fesl, J. (2023) "A Novel Dataset for Encrypted Virtual Private Network Traffic Analysis", Data in Brief.

Sowinski-Mydlarz, V., Li, J., Ouazzane, K., Vassilev, V. (2022) "Threat Intelligence Using Machine Learning Packet Dissection", London Metropolitan University Cyber Security Research Centre.

Lichy, Adi & Bader, Ofek & Dubin, R. & Dvir, Amit & Hajaj, Chen. (2022). When a RF Beats a CNN and GRU, Together -- A Comparison of Deep Learning and Classical Machine Learning Approaches for Encrypted Malware Traffic Classification.

de Brito Guimarães, L.C., 2023. DEEP LEARNING-BASED REAL-TIME BOTNET DETECTION FOR EDGE DEVICES (Doctoral dissertation, Universidade Federal do Rio de Janeiro).

Ndichu, s & Okoth, Sylvester & Okoyo, Henry & Wekesa, Cyrusd. (2023). Detecting Remote Access Network Attacks Using Supervised Machine Learning Methods. International Journal of Computer Network and Information Security. 15. 48-61.

N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 (MilCIS), Canberra, ACT, Australia, 2015, pp. 1-6.

Asswad, A., 2022. Analysis of attacks and prevention methods in cybersecurity (Doctoral dissertation, University of Brescia).

Patel, K.C. and Sharma, P., 2017. A Review paper on pfsense-an Open-source firewall introducing with different capabilities & customization.

Meena, A.K., Hubballi, N., Singh, Y., Bhatia, V. and Franke, K., 2020, December. Network Security Systems Log Analysis for Trends and Insights: A Case Study. In *2020 IEEE (ANTS)* (pp. 1-6).

OpenArgus, (2024). Documentation. Available at: https://openargus.org/documentation (Accessed: 5 April 2024)