www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 578-596. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.894

Analysis of PowerSHAP Feature Selection Method on APT Detection in Network Traffic Dataset

Adrián Hernández Rivas[1], *Victor Morales-Rocha*[1], *J. Patricia Sánchez Solis*[1,*]
[1] Universidad Autónoma de Ciudad Juárez, México.
adrian.rivas@uacj.mx, victor.morales@uacj.mx
*Corresponding autor: julia.sanchez@uacj.mx

Abstract. Novel feature selection methods are emerging to improve the accuracy of machine learning classifiers, including the method PowerSHAP (PS). This work investigates the efficacy of PS in enhancing Advanced Persistent Threat (APT) prediction performance across Random Forest, Decision Tree, and XGBoost classifiers. Experiments were conducted using the Dapt2020 and Unraveled network traffic datasets, both designed for APTs detection and containing diverse simulated attack scenarios. Performance evaluation metrics of experiments include accuracy, precision, recall, and F1-score. The findings contribute valuable insights into the application of PS feature selection for improving APT detection in complex network environments.

Keywords: APT, PowerSHAP, Cybersecurity, Network Traffic.

Article Info
Received February 01, 2025
Accepted April 02, 2025

1 Introduction

In the digital era, Advanced Persistent Threats have emerged as a significant concern for organizations, demanding adaptive and robust cyberdefense mechanisms (Friedberg et al., 2015). Modern cyberattacks are characterized by rapid evolution and escalating complexity, necessitating advanced cybersecurity strategies that integrate Artificial Intelligence (AI) to safeguard sensitive data and critical infrastructure. APTs are sophisticated, long-term attacks designed to infiltrate systems and discreetly exfiltrate confidential information (Friedberg et al., 2015). APTs frequently target organizations to access trade secrets, intellectual property, and classified data, underscoring their strategic and financial motivations (Rajendran et al., 2024).

The global market dedicated to mitigating Advanced Persistent Threats (APTs), which are characterized by their sophisticated and prolonged cyberattacks against organizations, was valued at $5.9 billion in 2021. Forecasts indicate a substantial expansion to $30.9 billion by 2030, representing an anticipated annual growth rate of 20.5% between 2022 and 2030. This projected surge underscores the escalating sophistication of cybercrime and the commensurate need for robust cybersecurity measures. As APTs become increasingly intricate and detrimental, organizations worldwide are significantly increasing their investments in defensive strategies to safeguard sensitive data, critical infrastructure, and organizational reputation. The predicted market value of $30.9 billion highlights the imperative for advanced cybersecurity technologies and strategies capable of effectively countering these clandestine and high-impact attacks, see Figure 1.

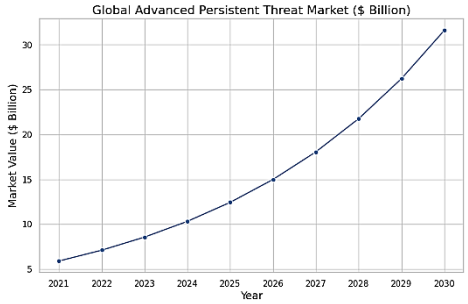

Figure 1. Projected Market Growth for APT Detection Solutions (2021-2030),

Network traffic analysis has emerged as a crucial approach for detecting Advanced Persistent Threats (APTs), with researchers exploring various methodologies and techniques. Multiple studies have demonstrated the effectiveness of machine learning approaches in APT detection. For instance, (Joloudari et al., 2020) found that deep learning models achieved superior performance with 98.85% accuracy and a low false positive rate of 1.13% compared to traditional machine learning methods. Supporting these findings, (Xuan et al., 2021) reported similarly high performance using Random Forest algorithms, achieving 97.56% accuracy for APT domain detection. The most commonly used machine learning classifiers for this type of problem, when using a supervised learning approach, include Decision Tree (DT), Naïve Bayes, eXtreme Gradient Boosting (XGB), Random Forest (RF), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM)(Al-Saraireh & others, 2022).

Several researchers have proposed specialized frameworks for APT detection. HOLMES, developed by (Milajerdi et al., 2018), offers real-time detection capabilities with high precision and low false alarm rates, while providing visual summaries of attacker actions to assist cyber response teams. Similarly, (Marchetti et al., 2016) developed an approach focused on identifying suspicious internal hosts in large networks with approximately 10,000 hosts, making it more feasible for security specialists to analyze potential threats. The integration of multiple detection layers and advanced analytics has shown promise. (Zimba et al., 2020) demonstrated that combining semi-supervised learning with complex network characteristics could achieve a 90.5% detection precision across different APT attack stages. (Wang et al., 2014), introduced an innovative "network gene" concept for characterizing behavior patterns in network applications, offering a new perspective on APT monitoring.

Recent developments have focused on improving detection accuracy through specialized techniques. (Dijk, 2021), introduced a novel method of analyzing payload data in network traffic flow, specifically addressing the challenging data exfiltration stage of APTs. Similarly, (Eke et al., 2019), explored LSTM-RNN models, achieving remarkably high accuracy (99.99%) in identifying attacks from normal network behavior. However, as (Alshamrani et al., 2019), suggests in their survey that significant challenges remain. The rapid evolution of APT attack tools and techniques continues to outpace existing security measures, and there is still a lack of comprehensive solutions that can detect APT cyberattacks from start to finish. This suggests that while considerable progress has been made in network traffic analysis for APT detection, further research is needed to develop more robust and adaptive detection systems. Network Traffic generates common standard features used in literature datasets, as shown in Figure 2, data from (Al-Saraireh et al., 2022).

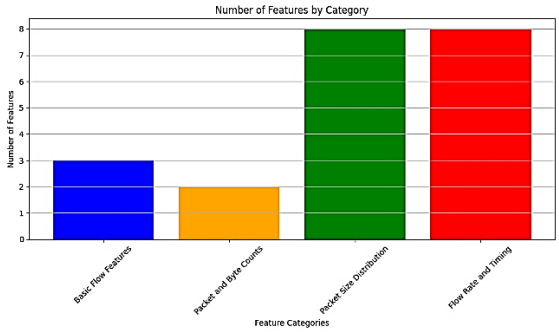

Figure 2. Network Traffic standard features.

The contributions of this research are described as follows:

1. Analysis of PowerSHAP impact to enhance classifier performance in APT detection over Network Traffic data.
2. To analyze the efficacy of feature selection Powershap in different datasets with imbalanced multiclass.

The following sections are organized as follows: Section 2 presents the related works on feature selection in network intrusion detection. Section 3 describes the research methodology for evaluating PowerSHAP, including experimental design and setup. Section 4 details the data acquisition and preprocessing techniques applied to the DAPT2020 and Unraveled datasets. Section 5 discusses the application and results of PowerSHAP. Section 6 explains the model training configurations and conditions. Section 7 presents the experimental results and analysis. Lastly, Section 8 provides the conclusion and outlines future work directions.

2 Related works on feature selection in network intrusion detection

Feature selection is the process of identifying and selecting a subset of the most relevant features (variables or predictors) from a dataset to construct effective predictive models. This dimensionality reduction technique aims to enhance model performance and improve interpretability. In the context of Network Intrusion Detection Systems (NIDS), feature selection is particularly crucial for the following reasons (Ahmed et al., 2024):

- Enhanced detection accuracy: By isolating the most significant attributes indicative of intrusion patterns, feature selection improves the precision and reliability of NIDS models.
- Improved model efficiency and reduced complexity: Eliminating irrelevant and redundant features mitigates noise, reduces computational overhead, and minimizes the risk of overfitting, leading to lower false positive rates and improved overall efficiency.

Several recent studies have explored feature selection techniques for network intrusion detection, with some specifically aiming to enhance the detection of Advanced Persistent Threats (APTs). A brief overview of these works is provided below.

Hofer-Schmitz et al. (2021) conducted a correlation analysis, incorporating a detailed investigation via boxplots to identify suitable features. These were then organized into distinct sets, and their impact on detection capabilities was evaluated using the Local Outlier Factor method. Al-Zoubi & Altaamneh (2022) proposed a wrapper-based feature selection approach utilizing the Chaotic Crow Search Algorithm (CCSA) for anomaly-based network intrusion detection systems. The effectiveness of this method was demonstrated using the LITNET-2020 dataset. Qi et al. (2023) developed an efficient feature selection algorithm that initially assesses feature correlation and pairwise redundancy relative to class labels, employing an enhanced Pearson correlation coefficient. Subsequently, they refined the evaluation function based on conditional mutual information to derive a final feature subset, aiming to improve classification rates and accuracy. Kumar et al. (2024) introduced

a fuzzy-inference-driven feature selection method combined with optimized deep learning. Their approach employed a Deep Convolutional Neural Network (DeepCNN) optimized by the Smart Flower Cosine Algorithm (SFCA). Feature selection was based on fuzzy-based distance measures, and data preprocessing involved quantile normalization and data augmentation. Rai et al. (2024) utilized a novel feature engineering methodology that included advanced feature scaling and Random Forest-based feature selection techniques. This work evaluates the performance of three traditional classification models—Naive Bayes (NB), Logistic Regression (LR), and Support Vector Classifier (SVC). Sakthivelu et al. (2024) explored various feature extraction techniques, including Analysis of Variance (ANOVA) F-test, Mutual Information, Recursive Feature Elimination, and Permutation Importance, to identify optimal features within the dataset. Sakthivelu & Kumar (2024) aimed to enhance APT detection by employing the Grey Wolf Optimizer (GWO) algorithm for feature selection. This approach mimics the social behavior and hunting strategies of grey wolves to identify the most significant features contributing to APT detection. Liu et al. (2025) combined Principal Component Analysis (PCA) with the Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) algorithm to compress network traffic features and extract high-correlation features from APT datasets.

3 Research methodology for evaluating PowerSHAP

This work evaluates the effectiveness of PS, a Shapley value-based feature selection method for anomaly detection, in improving classifier performance for Advanced Persistent Threat (APT) detection in network traffic. The methodology is structured into six experiments using the DAPT2020 and the Unraveled dataset, the first has a labeled collection of network traffic spanning five days (Monday to Friday), with APT attacks injected from Tuesday onward. The second, it emulates an organizational environment over a six-week period, highlighting the behavior of employees and the corresponding malicious activities from various attacker skill levels the dataset used in experiments corresponds to week 6, day 2. A general description of the methodology is described in Figure 3. The unraveled dataset incorporates advanced APTs in network traffic over weeks 2-6, for the experiment's dataset of Week 6-day 2 with the majority of advanced and complex APTs.

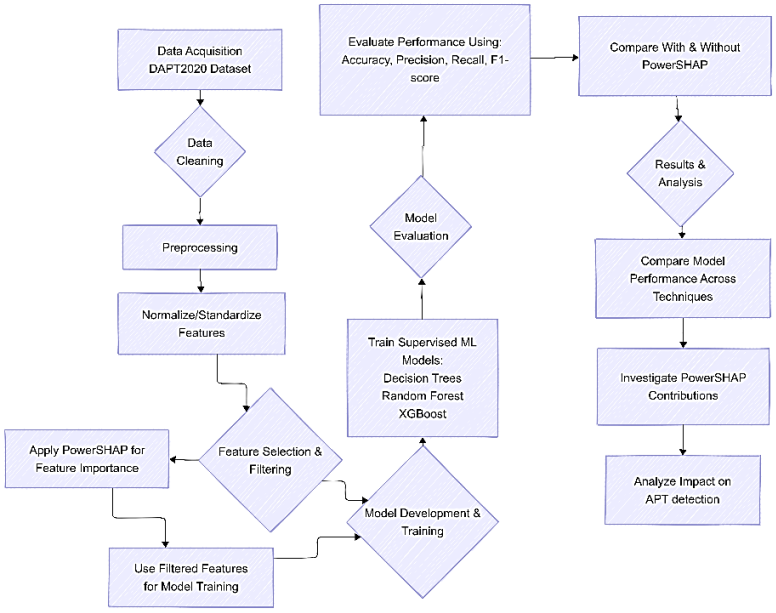

Figure 3. Outline of the Research Methodology Employed in This work.

3.1 Experimental Design and Setup

Experiments were implemented on the proprietary platform Colab (Google Inc., 2024), a cloud environment connected to cloud storage licensed by Google. As shown in Table 1, the computational environment provided utilizes a virtualized infrastructure based on two Intel Xeon processors, as evidenced by the consistent '*GenuineIntel*' vendor ID and *'Intel(R) Xeon(R) CPU@ 2.20GHz*' model name across both allocated CPU cores. Each core operates at a frequency of approximately 2.20 GHz, with a shared *L3* cache size of 56320 KB. The system is provisioned with 12.7 GB of RAM and 107.7 GB of disk space, facilitating the execution of computationally intensive tasks. The platform, accessible from any web browser, essentially offers Jupyter Notebooks hosted in the cloud. This eliminates the need for users to set up their own local environments. The free tier's resource allocation is subject to availability and runtime limitations, resulting in potential hardware variations and performance fluctuations. Seamless integration with cloud storage Google Drive enables efficient data management, highlighting Colab's design for scalable, accessible computational workflows.

Table 1. Specifications of experimental environment for classifier evaluation

Hardware specification	Description
CPU Vendor ID	GenuineIntel
CPU Family	6
CPU Model	79
CPU Model Name	Intel(R) Xeon(R) CPU @ 2.20GHz
CPU Stepping	0
CPU MHz	2,199.998
Cache Size	56,320 KB
RAM	12.7 GB
Storage	107.7 GB

The following six experiments were conducted to assess the impact of PowerSHAP (PS) on classifier performance:

- **Experiment 1:** The classifiers used in the first experiment were Random Forest (RF), Decision Tree (DT), and XGBoost (XGB) for APT prediction, using imbalanced data trained without oversampling dataset and without PS with classes on Wednesday Dapt2020. The dataset includes APTs such as *SQL Injection, Directory Bruteforce, Account Bruteforce, CSRF*, and *Account Bruteforce*. *Malware Download, Account Discovery.* Data are described in subsection 3.1. A summary of the APTs instances is shown in Table 2.

- **Experiment 2:** In the second experiment, the same classifiers were used over the imbalanced Wednesday dataset without oversampling but using PS.

- **Experiment 3:** In the third experiment, the same classifiers were used over the Wednesday dataset using oversampling and PS.

- **Experiment 4:** The classifiers used in the fourth experiment were Random Forest (RF), Decision Tree (DT), and XGBoost (XGB) for APT prediction, using imbalanced data trained without oversampling dataset and without PS with classes on Week 6 day 2 of Unraveled dataset; it includes APTs such as *Normal, Maintain Access, Bruteforce, Active Scanning, Encrypted Channel*, *Hijack Execution.* Data are described in subsection 3.2. A summary of unprocessed data instances for this dataset is shown in Table 3.

- **Experiment 5:** In the fifth experiment, the same classifiers were used over the imbalanced Week 6 day 2 Unraveled dataset without oversampling but using PS.

- **Experiment 6:** In the sixth experiment, the same classifiers were used over the Week 6 day 2 Unraveled dataset using oversampling and PS.

Classifiers evaluation protocol

- Classifiers: Decision Trees, Random Forest, and XGBoost (selected for their established efficacy in cybersecurity tasks).
- Metrics: F1-score, precision, recall, and accuracy.
- Cross Validation

Implementation details

PowerSHAP: Applied independently for each attack class to identify feature importance (see Table 3).

Methodology key features

Comparative Framework: Directly contrasts feature selection with PS against balanced and unbalanced data in classifier performance (Decision Tree, Random Forest, XGBoost).

4 Data Acquisition and Preprocessing Techniques

The experiments presented in this work use the DAPT2020 and Unraveled datasets, which are public network traffic datasets covering different APT stages. The data of each dataset was generated by a research group using a sandbox environment with real, virtualized infrastructure.

4.1 Dapt2020 Dataset Description

Dapt2020 is a dataset generated by the research group of Myneni et al. (2020). It was collected over five days, each corresponding to a distinct phase of an Advanced Persistent Threat (APT) attack. The sequential progression of attack stages allows for a comprehensive analysis of adversarial behavior in network environments.

- Day 1 (Monday): Normal network activity was recorded to establish a baseline for subsequent comparisons.
- Day 2 (Tuesday): The Reconnaissance phase was executed, utilizing tools such as NMap, Burp Suite, and Dirbuster to systematically identify vulnerabilities within systems and web applications.
- Day 3 (Wednesday): The attack advanced to the Establish Foothold phase, incorporating techniques such as SQL injection, Cross-Site Request Forgery (CSRF) attacks, and remote shell deployments to gain initial access and persistence.
- Day 4 (Thursday): The Lateral Movement phase ensued, characterized by privilege escalation, exploiting MySQL vulnerabilities, and unauthorized access to multiple networked systems.
- Day 5 (Friday): The attack culminated in Data Exfiltration, where sensitive files were illicitly transferred, accompanied by denial-of-service (DoS) attacks to disrupt normal operations.

The dataset encapsulates a wide range of attack vectors, including credential compromise, privilege escalation, and command injection. Feature extraction was conducted using CICFlowMeter (Network monitoring traffic tool), resulting in 85 attributes that comprehensively characterize network flow behavior. Given that the dataset includes benign and malicious traffic, it is a robust resource for machine learning-based intrusion detection research. Additionally, each record is meticulously labeled, providing granular insights into specific attack methodologies and their corresponding positions within the cyber kill chain framework. In this work, the dataset from Wednesday was used because it contains the majority of APT attack classes.

Dataset structure and feature descriptions

The dataset comprises 85 attributes, which can be categorized into four principal groups: network flow identification, traffic statistics, packet behavior, and attack classification. The following is a structured overview of key features:

- **Network flow identification.** These attributes define the fundamental properties of each network flow instance:
- **Traffic duration and rate metrics.** These attributes quantify the temporal characteristics and data transmission rate network flows:
- **Forward (Fwd) and backward (Bwd) packet characteristics**. This category describes packet dynamics in both the forward (originating) and backward (response) directions
- **TCP flag analysis.** These attributes capture the presence and frequency of Transmission Control Protocol (TCP) control flags, which indicate specific network communication behaviors
- **Packet size and flow statistics.** This set of features describes variations in packet sizes and network flow dynamics
- **Average packet size:** Mean packet size across the entire network flow.
- **Attack classification:** These columns provide labels for cybersecurity research.
  Activity. Description of the malicious or normal activity observed.
  Stage. The stage of the cyber kill chain the activity belongs to (e.g., Reconnaissance, Lateral Movement, and Data Exfiltration).

The Wednesday DAPT2020 dataset includes *Directory Bruteforce* (8,465 instances), *SQL Injection* (55 instances), *CSRF* (7 instances), *Account Discovery* (12 instances), *Account Bruteforce* (91 instances), *Malware Download* (2 instances), and *Normal traffic* (8,855 instances) as shown in Table 2 with 17,487 total instances. Class imbalance is evident, particularly in rare attacks (e.g., Backdoor, and CSRF). Cyberattacks such as Backdoors evading authentication and network scans probing for vulnerabilities via tools like Nmap are diverse. Directory brute-force attacks systematically expose hidden resources using automated tools (*e.g., DirBuster*), while malware delivery exploits vectors like phishing to enable data theft or disruption. Injection-based exploits, such as SQL manipulation or CSRF, target application layers to extract sensitive data or force unauthorized actions. Reconnaissance tactics, including account discovery and credential brute forcing, facilitate unauthorized access by exploiting weak authentication, underscoring the multifaceted nature of modern cyber threats (Shostack, 2014).

Table 2. Dapt2020 Wednesday dataset.

Label	Instances	Percentage
Directory Brute Force	8,465	48.41
Malware Download	2	0.01
SQLInjection	55	0.31
CSRF	7	0.04
Account Discovery	12	0.07
Account Bruteforce	91	0.52
Normal	8,855	50.64
Total	**17,487**	**100**

Standard preprocessing techniques on dataset Dapt2020:

1. **Data Loading**: Starts loading a dataset that contains network traffic data. This dataset is loaded into a Pandas DataFrame.
2. **Cleaning**: Drops irrelevant columns such as Flow ID, Src IP, Dst IP, Timestamp, and Stage. These columns are not relevant for the training of the machine learning model.

3. **Normalization**: The features are scaled using MinMaxScaler, which normalizes the data to a range of [0, 1]. This is important for algorithms sensitive to the input data's scale, such as XGBoost.
4. **Train-Test Split**: The dataset is split into training and testing sets using an 80-20 split ratio. This is a common practice to evaluate the performance of machine learning models.

4.2 Unraveled dataset description

The dataset was generated by Myneni et al. (2023), it is structured into two primary components: network traffic captures and host-level logs. The Unraveled dataset is a semi-synthetic cybersecurity resource designed to capture the behavior of Advanced Persistent Threats (APTs). It emulates a realistic organizational environment over a six-week period, highlighting the behavior of employees and the corresponding malicious activities from various attacker skill levels. In week 6, day 2, the dataset exhibits the highest APT activities, where attackers executed stealthy operations that blended seamlessly with normal user behavior. This specific day serves to illustrate the challenges faced in detecting sophisticated threats, emphasizing the need for advanced detection techniques that consider the complexities of real-world cyber-attacks. The Unraveled dataset includes several key features to categorize APT activities effectively.

The target column "Activity" in Week 6-Day2 with total of 253,922 instances, explicitly including attacks such as: *Bruteforce: Password Guessing, Active Scanning: Vulnerability Scanning, Active Scanning: Scanning IP Blocks, Hijack Execution Flow: Path Interception by PATH Environment Variable, Encrypted Channel: Symmetric Cryptography* and Normal traffic such as: *text/html, text/css, image/jpeg, image/png, application/x-javascript, application/x-chrome-extension, application/octet-stream,* and *application/ocsn-stream*, *NaN* and *Unknown* flows as shown in Table 3.

Table 3. Dataset Week 6-Day 2 activity instances.

Activity	Instances
Maintain Access	8,011
Normal	221,839
text/html	4
text/css	1
application/x-javascript	2
application/x-chrome-extension	27
application/octet-stream	81
application/ocsp-response	3
image/jpeg	9
image/png	1
Bruteforce: Password Guessing	22,650
Active Scanning: Vulnerability Scanning	1
Active Scanning: Scanning IP Blocks	1,124
Hijack Execution Flow: Path Interception by PATH Environment Variable	1
Encrypted Channel: Symmetric Cryptography	3
nan	148
Unknown	17
Total	**253,922**

Data for experiments in this work are focused on specific data sourced from Week 6-Day 2 with the highest number of APT activities. Figure 4 illustrates the instances frequency of each APT activity on Unraveled Dataset for Week 6 - Day 2.

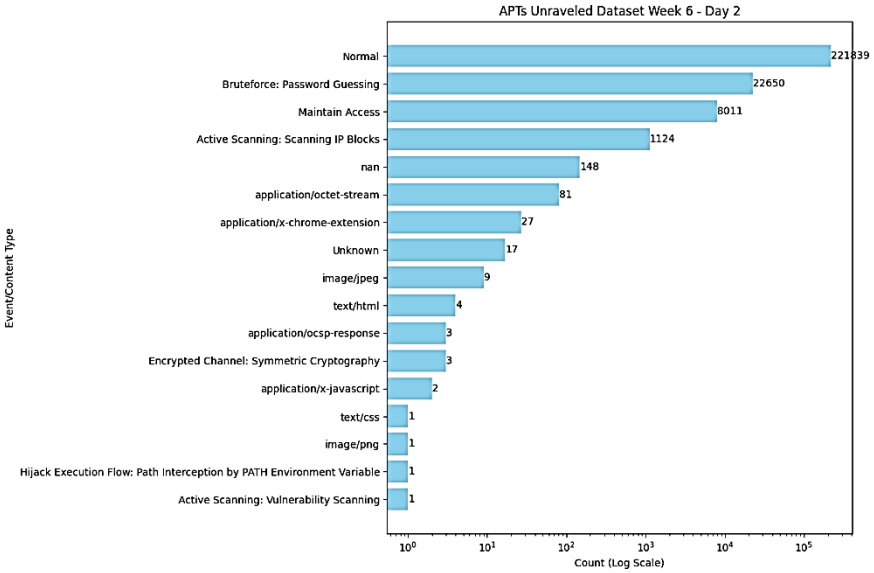

Figure 4. Unraveled activity label Week 6-Day 2.

The standard preprocessing on Unraveled Dataset Week 6-Day 2 is described as follows:

1. Data Loading: begin loading a dataset that contains network traffic data. This dataset is loaded into a Pandas DataFrame using the Colab platform in format CSV.
2. Data cleaning and preprocessing were performed to ensure consistency and data quality. The first activity involved the *'Hijack Execution'* class, which was used to generate one hundred new samples through controlled perturbation. This is a critical activity within the lifecycle of an Advanced Persistent Threat (APT), and the generation of synthetic samples expands the dataset. Furthermore, aggregation was applied to remove instances of other attack types, including *'Active Scanning'* and *'Normal'* activities, and those labeled *'nan'* and *'unknown'*. This was done because the focus of this work is to predict the type of attack being executed, without considering the phase of the attack lifecycle, defensive response, or signature. The resulting activity instances after preprocessing are described in Table 4.
3. Normalization: The features are scaled using MinMaxScaler, which normalizes the data to a range of [0, 1].
4. Train-Test Split: The dataset is split into training and testing sets using an 80-20 split ratio. This is a common practice to evaluate the performance of machine learning models.

Table 4. Synthesized (Week 6-Day 2) dataset.

Activity	Instances
Normal	221,967
Maintain Access	8,011
Bruteforce	22,650
Active Scanning	1,125
Encrypted Channel	3
Hijack Execution	101
Total Rows:	**253,857**

5 PowerSHAP

PowerSHAP feature selection method was proposed by Verhaeghe et al. (2022), and was employed in this work to quantify feature importance for APT detection in multiclass classification on the DAPT2020 and Unraveled datasets. PS, enhances machine learning feature importance quantification by integrating Shapley values from cooperative game theory with statistical hypothesis testing. It assigns a contribution score (Impact) to each feature and assesses its statistical significance using p-values. The p_value metric tests the null hypothesis (i.e., the feature's contribution is negligible). A low p-value (e.g., <0.01) rejects the null hypothesis, confirming the feature's significance measures the practical effect size with Cohen's ensures robust detection of true effects via statistical Power_0.01_alpha ensures rigorous control over false positives while maintaining sensitivity to detect true feature importance and iteratively increases trials for ≥99% power (99_Power_its_req) to minimize false negatives. Utilizing a Random Forest classifier, PowerSHAP provides interpretable and statistically reliable insights, particularly valuable for cybersecurity applications like APT detection, where precision and trustworthiness are critical.

5.1 Powershap Dapt2020 results

The results of the application of PS on Dapt2020 are illustrated in Table 5, main findings are presented and described as follows:

1. High-Impact Features: Attributes such as *Src-Port, Dst-Port, Flow-Duration, and Flow-IAT-Std* demonstrated significant impact and statistical relevance (*p_value = 0*), making them essential for accurate APT detection. These features consistently showed strong effect sizes and high statistical power, ensuring their reliability in distinguishing malicious activities.

2. Efficient Detection: Features like *Flow- IAT-Std and Packet- Length- Min required* fewer iterations to achieve 99% statistical power, highlighting their robustness and efficiency in contributing to the model's predictive performance.

3. Negligible Features: Attributes such as *Bwd-PSH-Flags, Fwd-Header-Length, and ACK- Flag- Count* had minimal impact (*p_value = 1, effect_size = 0*), indicating their limited relevance to the detection process. These features can be deprioritized in future analyses to streamline the model.

4. Statistical Rigor: The use of PowerSHAP ensured a robust evaluation of feature importance, combining Shapley values for interpretability with statistical metrics (e.g., effect size, power) for reliability. This approach minimized false positives and false negatives, enhancing the model's trustworthiness.

Table 5. PowerSHAP results on the Dapt2020 dataset.

.

Attribute	impact	p_value	effect_size	power_0.01_alpha	0.99_power_its_req
'Protocol',	0.006966	0	-3.755739	0.9999999999998803	4.602883612996893
'Flow Duration',	0.0069988	0	-4.9987094	1	3.848123144022605
'Src Port',	0.060986559838056564	0	-17.555489	1	2.4623291491879264
'Dst Port',	0.0680969	0	-14.446856	1	2.582001672882825
'Total Fwd Packet',	0.0038777	0	-2.9239433	0.9999999	5.6262562912145695
'Total Bwd packets',	0.0060233	0	-3.4840147	0.9999999999859839	4.864869583749631
'Total Length of Fwd Packet',	0.0024285	0	-3.7528539	0.9999999999998739	4.605402313480288
'Total Length of Bwd Packet',	0.0050413	0	-2.3087842	0.9999029036866228	7.132018793016248
'Fwd Packet Length Max',	0.0014495	0	-2.3836713	0.9999548841858588	6.886480300382986
'Fwd Packet Length Min',	0.005408	0	-2.4760274	0.9999832357204378	6.613324275653868
'Fwd Packet Length Mean',	0.0025415	0	-3.4962184	0.9999999999885815	4.851923845444724
'Fwd Packet Length Std',	0.0023209	0	-3.9728303	1	4.426983365797744
'Bwd Packet Length Max',	0.0025594	0	-3.8366143	0.9999999999999732	4.534301586317738
'Bwd Packet Length Min',	0.014044716954231262	0	-4.9473795	1	3.870132369399462
'Bwd Packet Length Mean',	0.0030654	0	-3.4409746	0.9999999999713158	4.9115209278175955
'Bwd Packet Length Std',	0.0012685	0.2	0	0	0
'Flow Bytes/s', 'Flow Packets/s',	0.0037349	0	-2.8310037	0.9999997641105691	5.794715778542293
'Flow IAT Mean',	0.0311239	0	-6.4237082	1	3.3979411043868124
'Flow IAT Std',	0.0365832	0	-9.5877296	1	2.9113118638268065
'Flow IAT Max',	0.010231074877083302	0	-4.4480513	1	4.117269846274375
'Flow IAT Min',	0.0084584	0	-4.7808921	1	3.945504223067063
'Fwd IAT Total',	0.006745	0	-3.8822957	0.9999999999999887	4.497214345201529
'Fwd IAT Mean',	0.0041665	0	-3.19446	0.9999999985844006	5.212407657574906
'Fwd IAT Std',	0.0029194	0	-3.8822088	0.9999999999999887	4.497283780159029
'Fwd IAT Max',	0.0021941	0	-2.0547796	0.998974	8.1722632
'Fwd IAT Min',	0.0023243	0	-2.1624387	0.9996046605591749	7.686714606009253
'Bwd IAT Total',	0.001714	0	-3.9684796	0.9999999999999979	4.4302631657038337
'Bwd IAT Mean',	0.0072995	0	-5.9085442	1	3.531825304352142
'Bwd IAT Std',	0.022510627284646034	0	-7.0720547	1	3.2601064407906932
'Bwd IAT Max',	0.0043698	0	-3.0529101	0.9999999886430397	5.4161433769672485
'Bwd IAT Min',	0.017787199467420578	0	-7.544342	1	3.1758773571177836
'Bwd PSH Flags',	1.1926169918297091e-06	1	0	0	0
'Fwd Header Length',	0	1	0	0	0
'Bwd Header Length',	0.0095492	0	-5.4518207	1	3.6750902243234527
'Fwd Packets/s',	0.0393931	0	-7.3275894	1	3.213080212766461
'Bwd Packets/s',	0.023402679711580276	0	-4.3506475	1	4.173789813240089
'Packet Length Min',	0.0360997	0	-11.767742	1	2.731211818781837
'Packet Length Max',	0.008557	0	-2.985925	0.9999999707674104	5.5220848777061972
'Packet Length Mean',	0.0025885	0	-3.0883374	0.9999999931830243	5.362704842396914
'Packet Length Std',	0.0027075	0	-2.5585282	0.9999933594946232	6.393383816939148
'Packet Length Variance',	0.0026103	0	-3.8371889	0.9999999999999736	4.5338279
'FIN Flag Count',	0.002723	0	-2.2904126	0.9998833970635436	7.195914694222329
'SYN Flag Count',	0.012081831693649292	0	-9.2847646	1	2.9429857998458773
'RST Flag Count',	0.0044647	0	-4.5760493	1	4.047546730874002
'PSH Flag Count',	0.0010644	0.1	0	0	0
'ACK Flag Count',	2.0204231532261474e-06	1	0	0	0
'Down/Up Ratio',	1.3787519037578022e-06	1	0	0	0
'Average Packet Size',	0.0053753	0	-4.2576728	1	4.230800330916116
'Fwd Segment Size Avg',	0.00253	0	-2.6523581	0.9999977918171027	6.167095711491578
'Bwd Segment Size Avg',	0	1	0	0	0
'Subflow Fwd Packets',	0	1	0	0	0
'Subflow Fwd Bytes',	0.0034561	0	-4.5233758	1	4.075643405352245
'Subflow Bwd Packets',	0.0027315	0.1	0	0	0
'Subflow Bwd Bytes',	0.0043094	0	-4.4120748	1	4.1377801707929445
'Bwd Init Win Bytes',	0	1	0	0	0
'Fwd Act Data Pkts',	0.0084514	0	-4.1592709	1	4.294686677124033
'Active Mean',	0.0022127	0	-3.1659088	0.9999999978255665	5.251470014804734
'Active Max',	0.0025175	0	-4.2678244	1	4.224424004475994
'Idle Mean',	0.0017866	0	-2.483633	0.9999845821569213	6.592150250385066
'Idle Max',	0.001694	0	-3.0765044	0.9999999919094509	5.3803631941841585

Visualization of PS results on Dapt2020 dataset is illustrated in Figure 5, findings are discussed in the following points:

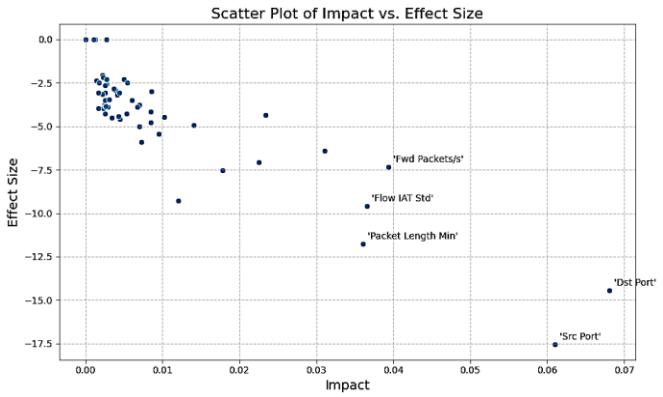

Figure 5. PowerSHAP on Dapt2020 dataset.

- Inverse Relationship: There appears to be a general inverse relationship between Impact and Effect Size. Features with higher Impact tend to have lower Effect Size, and vice versa. This suggests that

features with a significant contribution to the model's predictions might not necessarily have a large practical magnitude of influence when considered in isolation.

- Outliers: The points labeled *Dst Port* and *Src Port* are significant outliers, showing a high Impact but a very large negative Effect Size. This indicates that these features have a strong influence on the model's predictions but in a way that is substantially different from the average trend.

- Clustering: A cluster of points is observed in the lower left corner, indicating features with low Impact and moderate Effect Size. These features might have a smaller overall contribution but still exert a noticeable influence.

In summary, the findings underscore the importance of network flow attributes like port usage, packet timing, and flow duration in detecting APTs. By focusing on high-impact features and leveraging PowerSHAP's rigorous methodology.

5.2 PowerSHAP Unraveled results

The analysis of feature importance using PowerSHAP on the Unraveled dataset as shown in Table 6, revealed several critical insights for APT detection in multiclass classification. Key findings include:

- High-Impact Features: Attributes such as s*rc2dst-packets, src2dst-bytes, dst2src-first-seen-ms, and dst2src_last_seen_ms* demonstrated significant impact and statistical relevance (p_value = 0), making them essential for accurate APT detection. These features consistently showed strong effect sizes and high statistical power, ensuring their reliability in distinguishing malicious activities.

- Efficient Detection: Features like *src2dst-packets and dst2src-first-seen-ms* required fewer iterations to achieve 99% statistical power, highlighting their robustness and efficiency in contributing to the model's predictive performance.

- Negligible Features: Attributes such as *src-port, dst-port, ip-version, and vlan-id* had minimal impact (p_value = 1, effect_size = 0), indicating their limited relevance to the detection process. These features can be deprioritized in future analyses to streamline the model.

Table 6. PowerSHAP results on the Unraveled Week6-Day 2.

Attribute	impact	p_value	effect_size	power_0.01_alpha	0.99_power_its_req
src_port	9.705640877655242e-06	1	0	0	0
dst_port	8.201181117328815e-06	1	0	0	0
protocol	8.008610166143626e-05	0.5	0	0	0
ip_version	6.600494089070708e-07	1	0	0	0
vlan_id	4.7230107156792656e-05	0.875	0	0	0
tunnel_id	3.50420236827631e-06	1	0	0	0
bidirectional_first_seen_ms	3.3360367979184957e-06	1	0	0	0
bidirectional_last_seen_ms	2.032423253695015e-05	1	0	0	0
bidirectional_duration_ms	1.738812352414243e-05	1	0	0	0
bidirectional_packets	1.6695190424798056e-05	1	0	0	0
bidirectional_bytes	1.4284278222476132e-05	1	0	0	0
src2dst_first_seen_ms	1.4228849977371283e-05	1	0	0	0
src2dst_last_seen_ms	1.3444525393424556e-05	1	0	0	0
src2dst_duration_ms	1.0017349268309772e-05	1	0	0	0
src2dst_packets	0.015730392187833786	0	-11.1327	1	2.7765419498409973
src2dst_bytes	0.013024761341512203	0	-8.2791	1	3.065110826576835
dst2src_first_seen_ms	0.021153	0	-12.7666	1	2.668691
dst2src_last_seen_ms	0.021149	0	-8.73245	1	3.006474
dst2src_duration_ms	0.018961	0	-7.88361	1	3.1220204329447996
dst2src_packets	0.011895	0	-10.4961	1	2.827327
dst2src_bytes	0.009794	0	-5.92647	1	3.526724145454761
bidirectional_min_ps	0.008887	0	-5.24034	1	3.751419130418419
bidirectional_mean_ps	0.008745	0	-6.16647	1	3.461626166293254
bidirectional_stddev_ps	0.008555	0	-10.9034	1	2.7941618359702707
bidirectional_max_ps	0.008123	0	-9.08159	1	2.965420584946635
src2dst_min_ps	0.007798	0	-9.03925	1	2.9702255168887133
src2dst_mean_ps	0.007105	0	-8.95542	1	2.9798809324687547
src2dst_stddev_ps	0.007007	0	-5.6432	1	3.611777590165064
src2dst_max_ps	0.006446	0	-5.62305	1	3.6182070354107636
dst2src_min_ps	0.006158	0	-8.65356	1	3.0162211798459126
dst2src_mean_ps	0.005967	0	-9.44763	1	2.925702882178529
dst2src_stddev_ps	0.005901	0	-6.73608	1	3.327876517165385
dst2src_max_ps	0.00564	0	-8.20972	1	3.074679851493458
bidirectional_min_piat_ms	0.005455	0	-5.53853	1	3.645770904133746
bidirectional_mean_piat_ms	0.004687	0	-6.67986	1	3.3399532031652948
bidirectional_stddev_piat_ms	0.00456	0	-5.05803	1	3.8233542633409865
bidirectional_max_piat_ms	0.004554	0	-2.97758	1	5.535747940055678
src2dst_min_piat_ms	0.004161	0	-4.61023	1	4.029741129083525
src2dst_mean_piat_ms	0.004	0	-6.25621	1	3.4387423584515977
src2dst_stddev_piat_ms	0.003892	0	-4.45672	1	4.112389
src2dst_max_piat_ms	0.003877	0	-5.05395	1	3.825032752917185
dst2src_min_piat_ms	0.003868	0	-4.16672	1	4.289718084064991
dst2src_mean_piat_ms	0.003718	0	-5.53387	1	3.647318482466547
dst2src_stddev_piat_ms	0.003635	0	-5.35095	1	3.7106087338377503
dst2src_max_piat_ms	0.003348	0	-2.86239	0.9999999999999987	5.736082205992405
bidirectional_syn_packets	0.003275	0	-3.78589	1	4.576860405366269
bidirectional_cwr_packets	0.003231	0	-5.20669	1	3.7642434311624373
bidirectional_ece_packets	0.003108	0	-4.47848	1	4.100246356573745
bidirectional_urg_packets	0.003081	0	-7.55803	1	3.173604
bidirectional_ack_packets	0.003078	0	-4.51396	1	4.0807522757029355
bidirectional_psh_packets	0.003046	0	-6.89456	1	3.2950080727779785
bidirectional_rst_packets	0.002958	0	-5.58057	1	3.6319405256619546
bidirectional_fin_packets	0.002739	0	-5.9085	1	3.531839
src2dst_syn_packets	0.002655	0	-3.41878	1	4.936204696465661
src2dst_cwr_packets	0.002473	0	-5.1055	1	3.804028767772048
src2dst_ece_packets	0.002415	0	-3.398	1	4.959701623147888
src2dst_urg_packets	0.002288	0	-5.03409	1	3.8332638208433334
src2dst_ack_packets	0.002211	0	-2.72721	1	6.002518734779483
src2dst_psh_packets	0.00209	0	-4.27844	1	4.217790915801649
src2dst_rst_packets	0.001983	0	-4.00608	1	4.402224483978069
src2dst_fin_packets	0.001852	0	-4.40736	1	4.140501
dst2src_syn_packets	0.001838	0	-2.57832	0.9999999999975098	6.343647367437392
dst2src_cwr_packets	0.001804	0	-4.27983	1	4.216924105751325
dst2src_ece_packets	0.001759	0	-3.33759	1	5.030310972769013
dst2src_urg_packets	0.00174	0	-4.277	1	4.218688099074814
dst2src_ack_packets	0.001532	0	-4.33531	1	4.182982
dst2src_psh_packets	0.001464	0	-3.85833	1	4.516527734854275
dst2src_rst_packets	0.001337	0	-3.33944	1	5.028102025924469
dst2src_fin_packets	0.001331	0	-2.28103	0.9999999976895478	7.229145539107379
application_name	0.001292	0	-1.78487	0.99998	9.800782157168108
application_category_name	0.001284	0	-3.3408	1	5.026474095412504
application_is_guessed	0.000794	0	-1.41381	0.9973170065358236	13.763191613883118
requested_server_name	0.000663	0	-5.04305	1	3.829542778690341
client_fingerprint	0.000595	0	-4.82026	1	3.9271065772004254
server_fingerprint	0.000562	0	-1.96921	0.9999990262654794	8.616357753736345
user_agent	0.000377	0	-1.46924	0.9985683214562899	12.973358860550546
content_type	0.000342	0	-3.40802	1	4.948328792710207

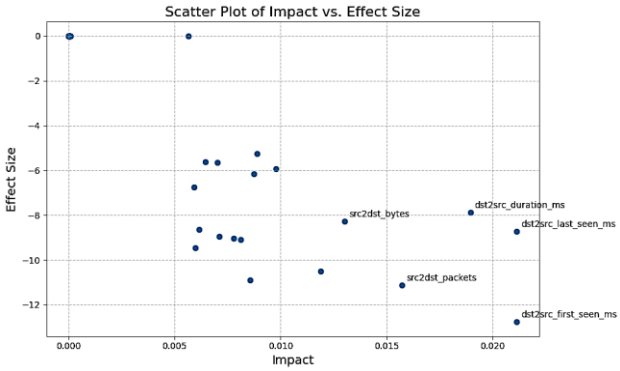

Figure 6. PowerSHAP on Unraveled dataset.

The observations and interpretation of Figure 6 are described as follows:

- Limited Range of Impact: The Impact values are clustered in a very narrow range (approximately 0 to 0.02). This suggests that all features have relatively low Impact scores, indicating that no single feature dominates the model's predictions.

- Wide Range of Effect Size: The Effect Size values span a much wider range, from approximately 0 to -12. This indicates that while the features might not have a substantial individual impact, they can have a significant magnitude of influence.

- Inverse Trend: There is a general trend of decreasing Effect Size as Impact increases. Features with slightly higher Impact tend to have more negative Effect Sizes.

Although the Impact values are low, the Effect Size values suggest that these features still play a role in the model's performance. The features with the most negative Effect Sizes (e.g., *dst2src-first-seen-ms*) might be particularly important for detecting specific patterns or anomalies. In summary, the findings underscore the importance of network flow attributes like packet counts, byte sizes, and timing metrics in detecting APTs

6 Model Training

Supervised machine learning models such as DT, RF, XGB were trained using the dataset: Wednesday DAPT2020 and Unraveled. Experimental conditions are described as shown in Table 7, Note that fields with value N/A it means that experiment not apply. Random Oversampling (ROS) was used for data balancing. The main objective of experiments is to analyze the influence of different combinations of previously highlighted techniques on the accuracy of classifiers.

Table 7. Model training conditions.

Algorithm	Dataset	Training-Test	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Decision Tree	Dapt2020	80/20	No PS + No ROS	PS + No ROS	PS + ROS	N/A	N/A	N/A
Random Forest	Dapt2020	80/20	No PS + No ROS	PS + No ROS	PS + ROS	N/A	N/A	N/A
XGBoost	Dapt2020	80/20	No PS + No ROS	PS + No ROS	PS + ROS	N/A	N/A	N/A
Decision Tree	Unraveled	80/20	N/A	N/A	N/A	No PS + No ROS	PS + No ROS	PS + ROS
Random Forest	Unraveled	80/20	N/A	N/A	N/A	No PS + No ROS	PS + No ROS	PS + ROS
XGBoost	Unraveled	80/20	N/A	N/A	N/A	No PS + No ROS	PS + No ROS	PS + ROS

7 Experimental results and analysis

The following subsections present the experimental results obtained from the application of the methodology to each dataset utilized in this study, specifically DAPT2020 and Unraveled.

7.1 Results from DAPT2020 dataset

This section describes the results of the experiments highlighted in the previous section. General metrics were used for the validation of the prediction for each model RF, DT, and XGBoost. PowerSHAP demonstrated varying effects across different classifiers:

The RF classifier exhibited a progressive improvement in performance across the three configurations. The mean *cross-validation* accuracy increased from 0.9921±0.0004 (No PS + No ROS) to 0.9989 (PS+ROS), demonstrating the beneficial impact of feature selection and oversampling. While the RF model achieved high precision, recall, and F1-scores (≥0.99) for majority classes like *Account Bruteforce* and *Malware Download* across all configurations, gains were more moderate for minority classes. Notably, the *Account Discovery* F1-score improved from 0.29 (PS+No ROS) to 0.77 (PS+ROS). These results suggest that while RF is robust for prevalent classes, oversampling enhances its ability to classify underrepresented attack vectors. The RF *Normal* class metrics improved with PS+ROS, but did not reach perfect scores. See Table 8.

Table 8. Random Forest results.

Class	Precision	Recall	F1-Score	Support
Normal	0.92	0.61	0.73	18
SQL Injection	0.00	0.00	0.00	3
Directory Bruteforce	0.00	0.00	0.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	0.99	1.00	0.99	1,771
Account Discovery	1.00	0.45	0.62	11
Accuracy	0.9934			3,502
Cross-validation accuracy .9928,.9917,.9921,.9917,.9921				
Mean cross-validation accuracy 0.9921				

a) Results No PS + No ROS

Class	Precision	Recall	F1-Score	Support
Normal	0.94	0.83	0.88	18
SQL Injection	1.00	0.33	0.50	3
Directory Bruteforce	0.00	0.00	0.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	0.99	0.99	0.99	1,771
Account Discovery	0.67	0.18	0.29	11
Accuracy	0.9923			3,502
Cross-validation accuracy .9946,.9921,.9935,.9914,.9903				
Mean cross-validation accuracy 0.9924				

b) Results PS + No ROS

Class	Precision	Recall	F1-Score	Support
Normal	0.93	0.72	0.81	18
SQL Injection	1.00	0.33	0.50	3
Directory Bruteforce	0.00	0.00	0.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	0.99	0.99	0.99	1,771
Account Discovery	0.67	0.91	0.77	11
Accuracy	0.992			3,502
Cross-validation accuracy .9984,.9988,.9991,.9987,.9992				
Mean cross-validation accuracy 0.9989				

c) Results PS + ROS

The Decision Tree classifier demonstrated the most substantial performance enhancements with the application of PS+ROS. Specifically, the *Account Discovery* F1-score increased from 0.91 (No PS + No ROS) to 0.96 (PS+ No ROS), highlighting the model's sensitivity to feature selection. For the *SQL Injection* class, F1-scores ranged from 0.5 (No PS+ No ROS) to 0.67 (PS+ No ROS), indicating improvement in the model's ability to classify this minority class SQL Injection. The *Normal* class metrics presented similar metrics in all experiments but did not reach perfect scores (See Table 9).

Table 9. Decision Tree results.

Class	Precision	Recall	F1-Score	Support
Normal	1.00	0.94.	0.97	18
SQL Injection	1.00	0.33	0.50	3
Directory Bruteforce	1.00	1.00	1.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	1.00	1.00	1.00	1,771
Account Discovery	0.91	0.91	0.91	11
Accuracy	0.9971			3,502
Cross-validation accuracy .9975,.9957,.9964,.9957,.9978				
Mean cross-validation accuracy 0.9966				

a) Results No PS+ No ROS

Class	Precision	Recall	F1-Score	Support
Normal	1.00	0.94	0.97	18
SQL Injection	0.67	0.67	0.67	3
Directory Bruteforce	1.00	1.00	1.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	0.71	1.00	0.83	5
Malware Download	1.00	1.00	1.00	1,771
Account Discovery	0.92	1.00	0.96	11
Accuracy	0.9963			3,502
Cross-validation accuracy .9985,.9975,.9964,.9975,.9932				
Mean cross-validation accuracy 0.9966				

b) Results PS+No ROS

Class	Precision	Recall	F1-Score	Support
Normal	0.95	1.00	0.97	18
SQL Injection	0.5	0.33	0.40	3
Directory Bruteforce	1.00	1.00	1.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	1.00	0.99	1.00	1,771
Account Discovery	0.67	0.91	0.77	11
Accuracy	0.996			3,502
Cross-validation accuracy .9990,.9988,.9989,.9994,.9993				
Mean cross-validation accuracy 0.9992				

c) Results PS+ROS

The XGBoost classifier achieved the highest overall performance and achieved perfect metrics (precision=1, recall=1, F1=1) for the *Normal* class with the application of PS+ROS, surpassing both RF and DT. Similar to the other classifiers, XGBoost demonstrated robust performance for the majority of classes across all configurations (See Table 10).

Table 10. XGBoost results.

Class	Precision	Recall	F1-Score	Support
Normal	1.00	0.94	0.97	18
SQL Injection	1.00	0.33	0.50	3
Directory Bruteforce	1.00	1.00	1.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	1.00	1.00	1.00	1,771
Account Discovery	0.91	0.91	0.91	11
Accuracy	0.9971			3,502
Cross-validation accuracy .9942,.9950,.9946,.9946,.9942				
Mean cross-validation accuracy 0.9946				

a) Results No PS+ No ROS

Class	Precision	Recall	F1-Score	Support
Normal	0.94	0.94	0.94	18
SQL Injection	1.00	0.33	0.50	3
Directory Bruteforce	0.00	0.00	0.00	1
Account Bruteforce	0.99	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	1.00	0.99	0.99	1,771
Account Discovery	0.64	0.82	0.72	11
Accuracy	0.9934			3,502
Cross-validation accuracy .9942,.9953,.9942,.9928,.9928				
Mean cross-validation accuracy 0.9939				

b) Results PS+No ROS

Class	Precision	Recall	F1-Score	Support
Normal	1.00	1.00	1.00	18
SQL Injection	0.33	0.33	0.33	3
Directory Bruteforce	0.00	0.00	0.00	1
Account Bruteforce	1.00	1.00	1.00	1,693
CSRF	1.00	1.00	1.00	5
Malware Download	1.00	1.00	1.00	1,771
Account Discovery	0.71	0.91	0.80	11
Accuracy	0.9951			3,502
Cross-validation accuracy .9990,.9992,.9988,.9991,.9993				
Mean cross-validation accuracy 0.9992				

c) PS+ROS

Across all classifiers, PS+ROS (PowerSHAP + Random Oversampler) consistently yielded the highest mean cross-validation accuracy 0.9989-0.9992, followed by PS+No ROS (0.9924-0.9966) and No PS+No ROS (0.9921±0.9966). However, classes with low support (e.g., *Directory Bruteforce*, *support=1*) presented challenges, resulting in unstable metrics (*precision/recall=0 or 1*). This underscores the necessity for additional data or advanced synthetic sampling techniques to address these limitations. Figure 7 illustrates the mean cross-validation accuracy for each algorithm across experiments when utilizing the Dapt2020 dataset.

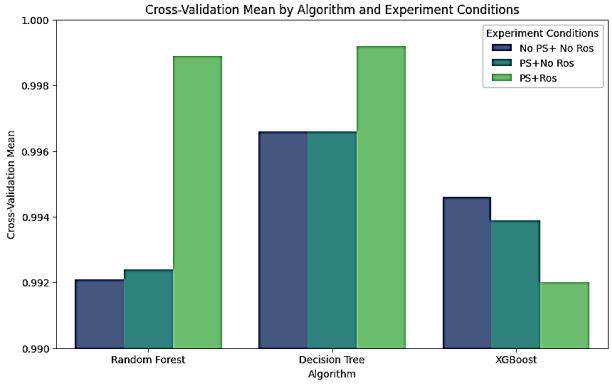

Figure 7. Cross validation mean results Dapt2020.

7.2 Results From Unraveled Dataset

Tables 11–13 show results for the Unraveled dataset for each classifier according to experiments 4–6 (defined in Table 7). RF model with the application of PS+No ROS achieved an accuracy of 0.999921216, with perfect precision, recall, and F1-scores (1.0) for all classes. Results across overall configurations are acceptable, except for the Minority class Encrypted Channel in configuration No PS+ No ROS (See Table 11).

Table 11. Random Forest Results Week 6-Day 2.

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	0.00	0.00	0.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.999921216			50,772
Cross-validation accuracy .9999, .9999, .9999, .9999, .9999				
Mean cross-validation accuracy 0.9998				

a) Results No PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.9999212164184984			50,772
Cross-validation accuracy .9999,.9999,.9999,.9999,1.0000				
Mean cross-validation accuracy 0.9999				

b) Results PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.9998818246277476			50,772
Cross-validation accuracy .9999,.9999,.9999,.9999,1.0000				
Mean cross-validation accuracy 0.99999				

c) Results PS + ROS

Decision Tree model with the application of PS+No ROS achieved an accuracy of 0.999802041, with perfect precision, recall, and F1-scores for all classes. Decision Tree model with PS+ROS similar results were achieved (See Table 12).

Table 12. Decision Tree Results Week 6-Day 2.

Class	Precision	Recall	F1-Score	Support
Active Scanning	0.99	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	0.00	0.00	0.00	1
Hijack Execution	0.95	1.00	0.98	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.999842433			50,772
Cross-validation accuracy .9995,.9998,.9997,.9997,.9997				
Mean cross-validation accuracy .9997				

a) Results No PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.999803041			50,772
Cross-validation accuracy .9996,.9998,.9999,.9999,.9999				
Mean cross-validation accuracy .9999				

b) Results PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	0.97	1.00	0.98	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.99978335			50,772
Cross-validation accuracy .9996,.9998,.9999,.9999,1				
Mean cross-validation accuracy .9998				

c) Results PS + ROS

XGBoost model with the application of PS+No ROS achieved an accuracy of 0.999803041, with perfect precision, recall, and F1-scores for all classes. This model achieved an accuracy of 0.99978335, with perfect precision, recall, and F1-scores for all classes with PS+ROS configuration. No PS+No ROS fail in classification of the minority class *Encrypted Channel* (See Table 13).

Table 13. XGB results Week 6-Day 2.

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	0.00	0.00	0.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.999842433			50,772
Cross-validation accuracy scores .9998,.9998,.9998,.9998,.09997				
Mean cross-validation accuracy .9998				

a) Results No PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	1.00	1.00	1.00	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.999803041			50,772
Cross-validation accuracy scores .9996,.9998,1,.9999,.9999				
Mean cross-validation accuracy .9998				

b) Results PS + No ROS

Class	Precision	Recall	F1-Score	Support
Active Scanning	0.99	1.00	0.99	225
Bruteforce	1.00	1.00	1.00	4,530
Encrypted Channel	1.00	1.00	1.00	1
Hijack Execution	1.00	1.00	1.00	20
Maintain Access	1.00	1.00	1.00	1,602
Normal	1.00	1.00	1.00	44,394
Accuracy	0.99978335			50,772
Cross-validation accuracy scores .9996,.9998,1,.9999,.09999				
Mean cross-validation accuracy .9998				

c) Results PS + ROS

The performance of various machine learning models was evaluated under different preprocessing conditions, including the use of PS feature selection and ROS. The results are presented in terms of precision, recall, F1-score, and accuracy for each class and cross-validation.
Figure 8 illustrates the mean cross-validation accuracy for each algorithm across experiments when utilizing the Unraveled dataset.

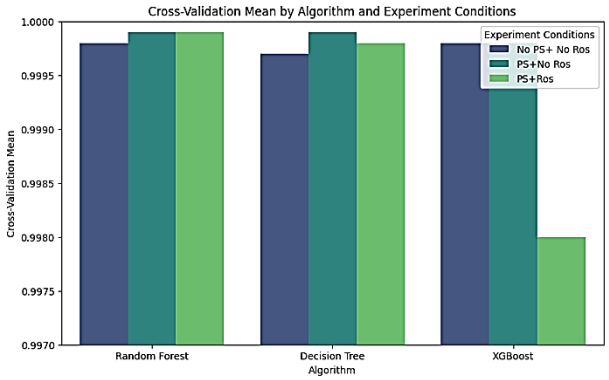

Figure 8. Cross Validation Results Unraveled dataset.

8 Conclusion and Future work

In conclusion, the integration of PowerSHAP feature selection with random oversampling enhanced the performance of all classifiers in Dapt2020 experiments, particularly for minority classes, while maintaining high accuracy. XGBoost and Decision Tree models exhibited the most substantial benefits from this configuration, suggesting its efficacy in handling imbalanced cybersecurity datasets.

Integrating PowerSHAP with balancing techniques showed significant promise in improving the detection of rare attack types without compromising overall accuracy. This hybrid approach combining PowerSHAP feature selection with appropriate balancing methods offers a viable strategy for optimizing intrusion detection systems, enabling enhanced detection capabilities across diverse attack types while maintaining high accuracy.

The results in the Unraveled dataset demonstrate that PowerSHAP used with and without Random Oversampling improves the performance of classifiers validated by metrics used in experiments. These findings underscore the potential of PowerSHAP as a valuable tool for improving the robustness and reliability of machine learning models in cybersecurity applications.

Future work is needed to evaluate alternative balancing techniques and classifiers through extensive experimentation, which will provide deeper insights into optimizing intrusion detection systems.

References

Ahmed, A., Asim, M., Ullah, I., Zainulabidin, & Ateya, A. A. (2024). An optimized ensemble model with advanced feature selection for network intrusion detection. *PeerJ Computer Science, 10*, e2472.

Al-Saraireh, J., et al. (2022). A novel approach for detecting advanced persistent threats. *Egyptian Informatics Journal, 23*(4), 45–55.

Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials, 21*(2), 1851–1877.

Al-Zoubi, H., & Altaamneh, S. (2022). A feature selection technique for network intrusion detection based on the chaotic crow search algorithm. In *Proceedings of the 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)* (pp. 54–60).

Dijk, A. (2021). Detection of advanced persistent threats using artificial intelligence for deep packet inspection. In *Proceedings of the 2021 IEEE International Conference on Big Data* (Big Data) (pp. 2092–2097).

Eke, H. N., Petrovski, A. V., & Ahriz, H. (2019). The use of machine learning algorithms for detecting advanced persistent threats. In *Proceedings of the 12th International Conference on Security of Information and Networks*. Recuperado de https://api.semanticscholar.org/CorpusID:208077048

Google Inc. (2024). *Google Colab* [Software]. https://colab.research.google.com/

Joloudari, J. H., Haderbadi, M., Mashmool, A., Ghasemigol, M., Band, S. S., & Mosavi, A. H. (2020). Early detection of the advanced persistent threat attack using performance analysis of deep learning. *IEEE Access, 8*, 186125–186137.

Kumar, A., Noliya, A., & Makani, R. (2024). Fuzzy inference based feature selection and optimized deep learning for advanced persistent threat attack detection. *International Journal of Adaptive Control and Signal Processing, 38*(2), 604–620.

Liu, G., Zhang, T., Dai, H., Cheng, X., & Yang, D. (2025). ResInceptNet-SA: A network traffic intrusion detection model fusing feature selection and balanced datasets. *Applied Sciences, 15*(2), 956.

Marchetti, M., Pierazzi, F., Colajanni, M., & Guido, A. (2016). Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks, 109*, 127–141.

Milajerdi, S. M., Gjomemo, R., Eshete, B., Sekar, R. C., & Venkatakrishnan, V. (2018). HOLMES: Real-time APT detection through correlation of suspicious information flows. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)* (pp. 1137–1152).

Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., & Kang, M. (2020). DAPT 2020—constructing a benchmark dataset for advanced persistent threats. In *Deployable Machine Learning for Security Defense: First International Workshop, MLHat 2020, San Diego, CA, USA, August 24, 2020* (Vol. 1, pp. 138–163).

Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A., & Huang, D. (2023). Unraveled—A semi-synthetic dataset for advanced persistent threats. *Computer Networks, 227*, 109688.

Qi, Z., Fei, J., Wang, J., & Li, X. (2023). An intrusion detection feature selection method based on improved mutual information. In *Proceedings of the 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*.

Rai, H. M., Yoo, J., & Agarwal, S. (2024). The improved network intrusion detection techniques using the feature engineering approach with boosting classifiers. *Mathematics, 12*(24), 3909.

Sakthivelu, & Kumar, V. (2024). Defence mechanism against advanced persistent threat attack using significant features based deep learning model. *Journal of Internet Services and Information Security, 14*(4), 263–277.

Sakthivelu, Kumar, V., Janaki, K., Ravi Kumar, D. V., Murali Krishna, C. V., Sravani, A., & Moses, G. J. (2024). Enhancing intrusion detection with advanced feature extraction through machine learning and deep learning methods. In *Proceedings of the 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)* (Vol. 14, No. 4, pp. 263–277).

Shostack, A. (2014). *Threat modeling: Designing for security*. John Wiley & Sons.

Verhaeghe, J., Van Der Donckt, J., Ongenae, F., & Van Hoecke, S. (2022). Powershap: A power-full Shapley feature selection method. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 71–87).

Wang, Y., Wang, Y., Liu, J., & Huang, Z. (2014). A network gene-based framework for detecting advanced persistent threats. In *Proceedings of the 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 97–102).

Xuan, C. D., Duong, D., & Hoang, X. D. (2021). A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic. *Journal of Intelligent & Fuzzy Systems, 40*, 11311–11329.

Zimba, A., Chen, H., Wang, Z., & Chishimba, M. (2020). Modeling and detection of the multi-stages of advanced persistent threats attacks based on semi-supervised learning and complex network characteristics. *Future Generation Computer Systems, 106*, 501–517.