



Novel Feature Selection Method for APT Detection

Hasanain M. J. Alfouadi^{1*}, Hiba Abdulrazzak Ahmed²

University of Al-Qadisiyah

*Correspondence: Hasanain M. J. Alfouadi

Email: hasanain.alfouadi@qu.edu.iq

Received: 28-04-2026

Accepted: 19-05-2026

Published: 06-06-2026



Copyright: © 2026 by the authors. Submitted for open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Abstract: An Advanced Persistent Threat (APT) is a multistage, highly sophisticated, and covert form of cyber threat that gains unauthorized access to networks to either steal valuable data or disrupt the targeted network. These threats often remain undetected for extended periods, emphasizing the critical need for early detection in networks to mitigate potential APT consequences. In this work, we propose a feature selection method for developing a lightweight intrusion detection system capable of effectively identifying APTs at the initial compromise stage. Our approach leverages the XGBoost algorithm and Explainable Artificial Intelligence (XAI), specifically utilizing the SHAP (SHapley Additive exPlanations) method for identifying the most relevant features of the initial compromise stage. The results of our proposed method showed the ability to reduce the selected features of the SCVIC-APT-2021, dataset, a benchmark dataset for APT detection that contains both benign and malicious traffic records, from 77 to just four while maintaining consistent evaluation metrics for the suggested system. The estimated metrics values are 97% precision, 100% recall, and a 98% F1 score. The proposed method not only aids in preventing successful APT consequences but also enhances understanding of APT behavior at early stages.

Keywords: APT; Cybersecurity; IDS; Feature Selection Techniques; SHAP.

Introduction

Intrusion Detection Systems (IDS) play a crucial role in cybersecurity by monitoring network traffic and identifying potential threats before they can cause significant damage. The importance of IDS is particularly pronounced in resource-constrained environments such as Internet of Things (IoT) networks, where interconnected devices create expansive and complex attack surfaces. These environments are characterized by limitations in processing power, memory, and energy sources, making the deployment of efficient and effective IDS a challenging task ([Zaman et al., 2021](#)). Within this context, Advanced Persistent Threats (APTs) represent a particularly insidious challenge. APTs are sophisticated, multi-stage attacks carried out by well-resourced adversaries who aim to remain undetected while exfiltrating sensitive information or causing operational disruption over extended periods ([Khaleefa & Abdulah, 2022](#)). The stealthy nature of APTs means that traditional detection methods often fail to identify them until significant damage has occurred. Given that APTs unfold in stages, early detection is critical as it allows for timely intervention and mitigation before an APT can fully execute its objectives ([Shaker et al., 2025](#)). However, the challenge lies in developing IDS that can effectively detect these elusive threats early, balancing the need for accuracy with the ability to process large volumes of network data efficiently.

In the realm of developing IDS based on machine learning (ML), the IDS is framed as a classification problem where the model is trained on a dataset to classify incoming

network traffic as either normal or malicious. By analyzing various features in the dataset, such as source and destination IP addresses, packet sizes, and protocol types, the ML model learns to distinguish between benign traffic and potential threats ([Santhosh Kumar et al., 2023](#)). These features are extracted using specialized tools like CICFlowMeter from the network traffic and converted into CSV files for further processing by the ML model during the training phase. However, not all of these features are relevant to identifying the type of traffic ([Kim & Pak, 2023](#)). Irrelevant features can adversely affect the model by increasing training time, system complexity, and reducing interpretability. This results in a lack of understanding of how the model makes predictions. Therefore, the process of selecting relevant features significantly influences the effectiveness of IDS ([Jaw, Ebrima and Wang, 2021](#)).

Three common methods for selecting relevant features are widely used in the ML domain. The first is a filter-based method, where the statistical properties of the data are used for determining the importance score for each feature in the dataset, such common methods are correlation or information gain. However, a drawback of this method is that it does not consider the impact of a specific ML algorithm.

The second method is wrapper methods, which utilize an ML algorithm in the feature selection process. These methods evaluate different feature subsets based on the model's performance and iteratively add or remove features to optimize the model. They are generally more accurate than filter methods but can be computationally expensive ([Wah et al., 2018](#)). Examples include forward selection, backward selection, and recursive feature elimination.

The last method is embedded feature selection, where the process of selecting relevant features is integrated directly into the model training process. Instead of selecting features separately before or after training, the model automatically determines which features are most important or relevant while learning from the data ([H. Liu et al., 2019](#)) ([Dhal & Azad, 2022](#)).

Our proposed detection model Light Weight-Intrusion Detection Sysytem LW-IDS, plays a crucial role in bridging the gap in recent IDS research by addressing key limitations. While most IDS research focuses on developing complex detection models using one or more ML algorithms or on extracting additional features to enhance the detection accuracy, our approach takes a different path. By minimizing the number of features while maintaining accuracy through SHAP-based feature selection within a lightweight IDS framework, We focus on achieving both efficiency and effectiveness, especially in IoT environments. This approach not only simplifies the detection process but also reduces computational overhead, making LW-IDS a practical and scalable solution for early-stage APT detection. As a result, it contributes to a more robust and scalable cybersecurity solution. The main contributions of this research are:

Explores the underutilized application of SHAP for feature selection in IDS, particularly in developing lightweight IDS for IoT environments.

Introduces a novel approach by integrating SHAP into a wrapper-based feature selection method, and optimizing IDS performance while balancing accuracy.

Develop a lightweight IDS capable of detecting APTs in their early stages, providing a scalable and cost-effective security solution.

The implications of our research are far-reaching, particularly in the context of IoT environments where resource constraints are a significant challenge. Developed LW-IDS can be integrated into various IoT devices and networks, providing robust security without compromising performance.

The rest of the paper is organized as follows: Section 2 presents related works in the field of APT detection. The proposed feature selection method is detailed in Section 3. Section 4 presents the results and compares the performance of the proposed method with other techniques. Finally, Section 5 concludes the paper with a summary of contributions, potential limitations, and suggestions for future work.

Related Works

In this section, we provide an overview of current studies related to APT detection, identifying their limitations, along with the ML algorithms used and the datasets employed, as summarized in Table I.

Tero and Timo ([Bodström & Hämäläinen, 2019](#)) proposed a deep learning stack that employs sequential neural networks for detecting APTs. The stack model consists of multiple detection algorithms rather than relying on a single algorithm. The model begins by detecting known attacks using multiple neural networks, which are then removed from the flow. In the second layer, normal traffic is filtered out after the detection process. The third layer, comprised of hybrid neural networks—specifically, Recurrent Neural Network-Long Short-Term Memory (RNN-LSTM) is utilized for detecting outlier samples, which are then removed in the fourth layer of the stack. However, the model suffers from high complexity due to the use of multiple detection algorithms.

In ([Xuan et al., 2021](#)), Xuan developed an early detection system for APTs by analyzing network traffic behavior using the random forest (RF) algorithm. The analysis began by examining and extracting traffic behavior based on the domain names of the IP addresses. Subsequently, these behaviors were transformed into features used to classify the traffic as either normal or anomalous, particularly focusing on traffic generated from the command-and-control server. To train and test the method, the author created a dataset by compiling 61 traffic files for APTs from various sources. Although the study's results demonstrate that the classifier can detect anomalous traffic with an accuracy of 99.9%, it is noteworthy that the collected data lacks a real representation of APT stages.

Hofer-Schmitz et al. in ([Hofer-Schmitz et al., 2021](#)) utilized various statistical network traffic features to generate different feature sets for detecting APT attacks. Two datasets, consisting of raw network data in PCAP format, were used for this purpose. The CICIDS2017 dataset was used as a resource of benign traffic and combined with malicious traffic extracted from the Contagio malware database ([Parkour, n.d.](#)). After feature extraction by CICFlowMeter tool, a feature selection method was applied using correlation analysis and boxplots. The CICFlowMeter tool was employed to extract features from the PCAP file of the dataset. Subsequently, an unsupervised method for detecting attacks was

proposed, employing the Local Outlier Factor (LOF) algorithm. The LOF algorithm calculates a score to measure the abnormality of each instance in the dataset based on its neighbors. Two evaluation metrics recall and true negative rate were employed to assess the performance of the feature sets. The highest weighted recall achieved was 75.8%, with a corresponding true negative rate of 90%. However, the work did not take the stages of APT into consideration.

In ([Shen et al., 2022](#)), Shen et al. proposed a model called Prior Knowledge Input (PKI), in this model, the dataset SCVIC-APT-2021 was re-classified by using clustering approaches to gain prior knowledge that is more suitable to combine with the features of the dataset. The output is subsequently processed using filter-based feature selection algorithms to identify the most influential features. These selected features were then fed into supervised models, specifically RF and XGBoost. The experimental results reveal that the RF classifier achieves a macro F1-score of 80.34% with 52 features, while XGBoost achieves 80.92% with 49 features. Overall, the proposed method aims to reduce the complexity of the supervised model. However, it simultaneously increases the model's complexity by incorporating an unsupervised model.

Ahmed and Sara in ([Alsanad & Altuwajjri, 2022](#)), used a set of unsupervised algorithms such as Hunt's Algorithm, K-Means, and Support Vector Machine (SVM), to develop a proposed framework for detecting APT. To prepare the CSE-CIC-IDS2018 dataset for clustering algorithms, the authors implemented multiple phases, including transformation, normalization, and feature selection based on principal component analysis. The primary finding of the experiment indicates that the SVM with Radial Basis Function (RBF) achieved a higher accuracy of 99.2% compared to other algorithms. It is noteworthy that the research used three clusters, which may not accurately reflect the number of APTs stages.

AL-Aamri et al. in ([AL-Aamri et al., 2023](#)), proposed Composition-Based Decision Tree (CDT) system for detecting APTs in live traffic by employing expert systems and ML algorithms. The study depends on the changes in the traffic flow for identifying anomalous traffic. The process involves various steps, such as linear regression analysis, feature extraction, and selection. To evaluate the proposed system, the authors combined different datasets with data generated from local frameworks. Anomalous traffic is manually labeled by domain experts with expertise in APTs. A decision tree-like model was created and fed into the naïve Bayes algorithm to generate a decision rule for APTs detection. The results of the CDT system show a precision of 96% in detecting malicious attacks. The average precision estimate for the proposed model was 94.3%. However, the study did not conduct multi-attack detections and focused on binary classification between normal and anomalous traffic.

Md Mahadi et al. in ([Hasan et al., 2023](#)), examined various boosting algorithms for identifying APTs using XAI. The SCVIC-APT-2021 dataset was used to evaluate these algorithms. The authors compared the performance of different boosting algorithms and observed that CatBoost and XGBoost outperformed other algorithms, achieving weighted F1-scores of 0.99 and 0.97, respectively. Although the paper employed XAI to analyze the

influence of individual features on the output to ensure model fairness and transparency, it did not exploit XAI to enhance the model's performance.

Zhu et al. in (Zhu et al., 2024), proposed a distributed framework for detecting APTs in IoT networks, named Global Vision Federated Learning (GV-FL). In this framework, a logistic regression model is distributed among multiple IoT devices for training on their local data. Subsequently, the parameters of each model are aggregated on a global server, which then redistributes them to the devices. The results of the framework achieved an accuracy of 92%. While the framework successfully preserves the privacy of IoT data, it does require additional time for sharing parameters between the IoT devices and the aggregate server.

Table I: summarized previous studies

Ref	Methodology	Dataset	Model	Limitations
(Bodström & Hämäläinen, 2019)	Utilizing a deep learning stack composed of multiple detection algorithms instead of relying on a single algorithm.	Privet dataset	RNN, LSTM	High complexity due to the use of multiple detection algorithms
(Xuan et al., 2021)	Analyzing traffic behavior to normal or anomaly traffic based on IP addresses	Privet dataset	RF	There is no APT stages
(Hofer-Schmitz et al., 2021)	Utilizing unsupervised learning to detect anomaly instances in the dataset	DARPA, CICIDS2017	LOF	The method doesn't detect individual APT attacks
(Shen et al., 2022)	Using clustering approaches to obtain prior knowledge for exploitation in classification algorithms to detect attacks	SCVIC-APT-2021	RF, XGBoost	Higher model complexity due to incorporating both unsupervised and supervised models.
(Alsanad & Altuwaijri, 2022)	Proposed a framework for detecting APTs using various unsupervised algorithms	CSE-CIC-IDS2018	SVM	No accurate representation of APT stages.
(AL-Aamri et al., 2023)	The study identifies anomalous traffic by demonstrating high sensitivity to changes in the live traffic flow of APTs.	Privet dataset	Decision Tree	There is no multi-attack detections.
(Hasan et al., 2023)	Examined multiple boosting algorithms for identifying APTs using XAI.	SCVIC-APT-2021	CatBoost, XGBoost	XAI is not used to enhance the model's performance.
(Zhu et al., 2024)	Developed a federated learning model that distributed among multiple IoT devices for training on their local data	Privet dataset	logistic regression	The method requires additional time for sharing parameters between the IoT devices and the aggregate server.

To explain the limitations of related works and how the proposed method addresses them, key challenges in existing research can be highlighted. Many IDS in these works face significant limitations, particularly in terms of model complexity. This complexity often arises due to the use of multiple detection algorithms or the combination of different learning approaches aimed at extracting more relevant features or identifying low-level patterns in the data. This complexity makes these models less suitable for deployment in resource-constrained environments like IoT networks. Another limitation is that these models that are used for detecting APTs often do not account for the multistage nature of such attacks. Instead, they typically focus on distinguishing between normal traffic and APT traffic, which means these systems may only identify an APT after it has progressed through multiple stages. This delayed detection reduces the effectiveness of preventative measures, as the APT might have already caused significant damage by the time it is identified. In this study, the limitations are addressed by developing a lightweight IDS (LW-IDS) with reduced computational complexity. This is achieved by selecting only the most relevant features and utilizing a multi-stage dataset, which enables LW-IDS to detect APTs at an early stage.

Methodology

The principal components of our proposed methodology for detecting APT at the I.C stage are organized into two distinct levels as depicted in Figure (1).

First, the methodology starts by preparing the dataset, which contains traffic flows before being fed into the prediction model. Next, the importance score for each feature is computed and used as the basis for the feature selection process. The lower-level components of the proposed method are explained in the following three steps:

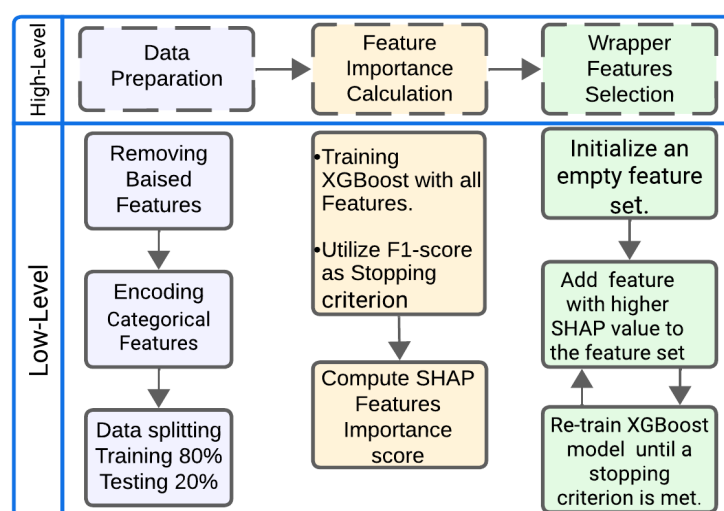


Figure. 1. Main Proposed Method Components

Step 1: Dataset Preparation

Several preprocessing steps are applied to make the dataset suitable for the training process. The process begins by removing biased features from the dataset to ensure the model is not unfairly influenced by attributes that could skew the results. Next, only the

two relevant stages of the dataset are isolated to focus on the specific stages of interest and eliminate unrelated data. Samples with missing values are then removed to maintain the integrity of the dataset and prevent incomplete information from affecting the analysis. Finally, categorical features are encoded, converting them into a numerical format suitable for ML models, enabling the algorithms to process and learn from these features effectively.

In this study, the SCVIC-APT-2021 dataset is used, which comprises 84 features extracted via the CICFlowMeter tool from the PCAP file representing network traffic ([J. Liu et al., 2022](#)). The reason for choosing this dataset is that all stages of APT are represented. Each stage contains various types of attacks. The I.C stage is exemplified by the 'smiley face' backdoor attack in Very Secure FTP Daemon (VSFTPD) version 2.3.4. This vulnerability allows users to log in using ':' and gain access to a command shell on port 6200 ([Koo et al., 2019](#)).

While the dataset comprises six stages, and since we aim to distinguish malicious I.C traffic from normal traffic, two sets of samples were isolated: Normal Traffic with 307,817 samples and I.C with 150 samples. Next, the features (flow id, time stamp, source IP, destination IP, source port, and destination port) were removed to prevent bias in predictions by the model and improve generalization for unseen data (Shen et al., 2022). Only 77 out of 84 features are used in the training process. Then, categorical features were converted to numerical values using a label encoder technique to make them compatible with ML algorithms that require numerical input ([Srikanth Yadav & Kalpana, 2019](#)). Subsequently, the pre-processed data is divided into two sets: an 80% training set, which is used to train the model on the relationships between the input features and the target variable, and a 20% testing set that is used to evaluate the model performance on new, unseen data. This simulates how the model will perform in the real world when it encounters data that was not part of the training process.

Step 2: Feature Importance Score Calculation

In this section, We determined each feature's importance score by using a cooperative approach between the trainable XGBoost model and SHAP. This method is crucial for the proposed LW-IDS as it provides a clear understanding of each feature's contribution to the model's predictions.

To compute feature importance scores, we utilized the TreeExplainer method within SHAP, which is designed for tree-based models ([Sharma et al., 2020](#)). The process starts by training the XGBoost model with all 77 features. The XGBoost model is configured with the following hyperparameters: a learning rate of 0.3, a maximum tree depth of 6, and 100 boosting rounds. The F1-score metric, which reflects the harmonic balance between precision (how accurately the model detects the I.C samples) and recall (the model's ability to detect the actual I.C samples), is used as a stopping criterion in the subsequent steps.

Next, important scores are identified that help define the influential features of a model's predictions. This is done by calculating the average of the absolute SHAP values for each feature across all samples in the dataset. Features with higher average absolute SHAP values are generally considered to have a greater impact on the model's predictions.

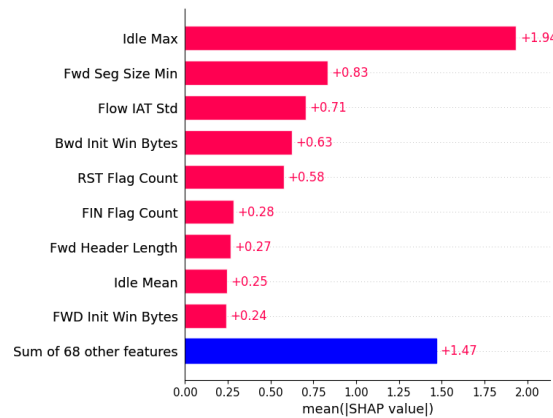


Figure 2. Top Nine features Importance Scores.

Figure (2) shows the importance scores for the top nine features.

Step3: Proposed Feature Selection

In this step, the relevant features are selected by using wrapper feature selection approaches. The process begins by creating an empty set of features and iteratively adding one feature at a time with a higher mean SHAP value to the set. The process continues until a predefined stopping criterion is met; if so, then the set contains the most relevant features with a direct impact on model predictions.

Result and Discussion

In this section, the results obtained from evaluating the LW-IDS, which was built using the proposed feature selection method, are presented. Furthermore, these results are compared with those from other feature selection methods and previous works

Performance Metrics

Precision, recall, and F1-score, which are the well-known metrics are used for assessing the performance of machine learning models. Precision measures how many of the positive predictions made by LW-IDS were correct, while recall measures how many of the I.C samples were correctly identified by the proposed model. F1-score strikes a balance between precision and recall, offering a unified measure of model performance. The mathematical formula for finding precision, recall, and F1-score are declare in (1), (2) and (3):

$$precision = TP / (TP + FP) \quad (1)$$

$$Recall = TP / (TP + FN) \quad (2)$$

$$F1 - score = 2 * (precision * Recall) / (precision + Recall) \quad (3)$$

Where:

TP: True Positives (correctly identified I.C samples).

FN: False Negatives (missed I.C samples).

FP: False Positive (incorrectly identified normal sample as I.C sample).

Experimental Results

Figure (3) shows the impact of the proposed feature selection on the performance of the XGBoost model was evaluated for a binary classification problem. The first case represents the model trained using all features in the dataset (77 features). This resulted in an F1-score of 98% and 97%, 100% for precision and recall respectively. The F1-score value is considered a stopping criterion for determining the best number of influential features. Then, we begin with an empty feature set and iteratively train the model using the feature that has the highest mean absolute SHAP value. This approach guarantees that features are added based on their importance, as indicated by their SHAP values in Figure (2), continuing until we reach the stopping criterion for feature selection. The optimal number of features for achieving the best results is four ('Idle Max', 'Fwd Seg Size Min', 'Flow IAT Std', and 'Bwd Init Win Bytes'). To further validate our findings, we can observe that adding the fifth most important feature ('RST Flag Count') does not improve the model's performance.

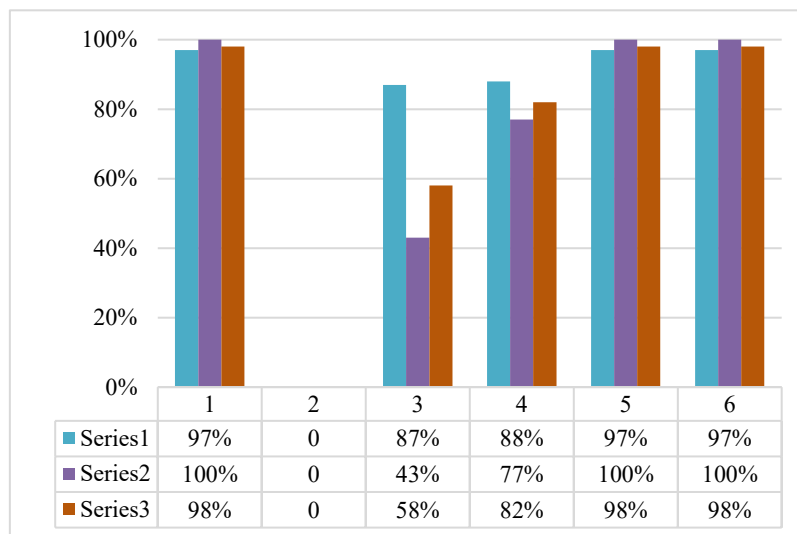


Figure 3. Performance Metrics for Different Feature Sets

Also, we compare the overall performance of our proposed method against various filter-based feature selection methods, such as Chi-squared and ANOVA, and the embedded method for the XGBoost classifier. Table II shows that our proposed method outperforms the other methods for the same number of features. This superiority stems from the fact that filter methods (Chi-squared, ANOVA, Mutual Information, and Pearson Correlation Coefficient), being statistical in nature, often rely on statistical tests that assume features are independent. However, in real-world datasets, features often exhibit complex relationships that filter methods might overlook. Conversely, wrapper methods consider all

possible feature combinations and can identify important features only in conjunction with others.

Table II. Comparative results with benchmark feature selection methods

Method	Top 4 Features	Pre	Rec	F1-score
Chi 2	'Total Length of Bwd Packet', 'Packet Length Std', 'Packet Length Variance', 'Subflow Bwd Bytes'	96%	68%	80%
ANOVA	'Total Length of Bwd Packet', 'Fwd Header Length', 'Packet Length Variance', 'Bwd Bytes/Bulk Avg'	90%	68%	78%
Mutual Information	'Idle Max', 'Idle Min', 'Idle mean', 'Protocol'	80%	62%	67%
Pearson Correlation Coefficient	'Packet Length Variance', 'Bwd Bytes/Bulk Avg', 'Total Length of Bwd Packet', 'Fwd Header Length'	95%	84%	89%
Embedded	'Bwd Packet Length Max', 'Total Length of Bwd Packet', 'Idle Std', 'Flow bytes'	100%	50%	67%
Proposed Method	'Idle Max', 'Fwd Seg Size Min', 'Flow IAT Std', 'Bwd Init Win Bytes'	97%	100%	98%

Furthermore, we compare the obtained results with similar works, such as (J. Liu et al., 2022) and (Shen et al., 2022), where the XGBoost algorithm was applied to the SCVIC-APT-2021 dataset. The study in (J. Liu et al., 2022) utilized 9 features and achieved an F1-score of 86%. Meanwhile, a filter-based feature selection method in (Shen et al., 2022) was used to select 49 features, resulting in an F1-score of 80%. In contrast, the proposed wrapper method, which relied on SHAP feature importance values, involved a more refined feature selection process, reducing the number of features to 4 and significantly improving the F1-score to 98%. The reason behind this is that the wrapper method using SHAP focuses on features that have the highest impact on the model, reducing the number of irrelevant or redundant features. This helps achieve better generalization and avoid overfitting.

Conclusion

This paper presents a novel method for selecting the most relevant features to develop novel lightweight IDS (LW-IDS) capable of detecting APTs at the initial compromise stage. The method is based on the XGBoost algorithm, which employs ensemble learning principles. The feature selection process uses XAI techniques, specifically the SHAP method, which utilizes SHAP values to determine the magnitude of feature importance. Utilizing these feature importance scores, the wrapper feature selection method identifies the most relevant features for APT detection. The proposed method successfully reduced the influential features to 4 out of 77, while the LW-IDS preserves the F1-score of 98%. This shows the effectiveness of our approach in achieving accurate APT detection with a smaller feature set, paving the way for the development of more streamlined and effective IDS solutions. In future work, we aim to address the class imbalance problem—an important issue in this dataset since minority attack classes are

underrepresented and can lead to biased detection results—by generating synthetic samples for the minority class using the Synthetic Minority Over-sampling Technique (SMOTE).

References

- AL-Aamri, A. S., Abdulghafor, R., Turaev, S., Al-Shaikhli, I., Zeki, A., & Talib, S. (2023). Machine Learning for APT Detection. *Sustainability (Switzerland)*, 15(18).
- Alsanad, A., & Altuwaijri, S. (2022). Advanced Persistent Threat Attack Detection using Clustering Algorithms. *International Journal of Advanced Computer Science and Applications*, 13(9), 640–649.
- Bodström, T., & Hämäläinen, T. (2019). A novel deep learning stack for APT detection. *Applied Sciences*, 9(6).
- Dhal, P., & Azad, C. (2022). A comprehensive survey on feature selection in the various fields of machine learning. In *Applied Intelligence (Vol. 52, Issue 4)*. Applied Intelligence. <https://doi.org/10.1007/s10489-021-02550-9>
- Hasan, M. M., Islam, M. U., & Uddin, J. (2023). Advanced Persistent Threat Identification with Boosting and Explainable AI. *SN Computer Science*, 4(3), 1–9.
- Hofer-Schmitz, K., Kleb, U., & Stojanović, B. (2021). The influences of feature sets on the detection of advanced persistent threats. *Electronics (Switzerland)*, 10(6), 1–22. <https://doi.org/10.3390/electronics10060704>
- Jaw, Ebrima and Wang, X. (2021). Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach. *Symmetry*, 13, 1764.
- Khaleefa, E. J., & Abdulah, D. A. (2022). Concept and difficulties of advanced persistent threats (APT): Survey. *Int. J. Nonlinear Anal. Appl*, 13(1), 2008–6822. <http://dx.doi.org/10.22075/ijnaa.2022.6230>
- Kim, T., & Pak, W. (2023). Integrated Feature-Based Network Intrusion Detection System Using Incremental Feature Generation. *Electronics*, 12, 1657.
- Koo, H., Ghavamnia, S., & Polychronakis, M. (2019). Configuration-driven software debloating. *Proceedings of the 12th European Workshop on Systems Security*, 1–6.
- Liu, H., Zhou, M., & Liu, Q. (2019). An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, 6(3), 703–715.
- Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., Bagheri, M., & Djukic, P. (2022). A new realistic benchmark for advanced persistent threats in network traffic. *IEEE Networking Letters*, 4(3), 162–166.
- Parkour, M. (n.d.). malware dump. Retrieved February 1, 2024, from contagiodump.blogspot.com
- Santhosh Kumar, S. V. N., Selvi, M., & Kannan, A. (2023). A Comprehensive Survey on Machine Learning-Based Intrusion Detection Systems for Secure Communication in Internet of Things. *Computational Intelligence and Neuroscience*, 2023, 1–24. <https://doi.org/10.1155/2023/8981988>
- Shaker, B. N., Al-Musawi, B., & Hassan, M. F. (2025). Explainable AI for enhancing IDS against advanced persistent kill chain. *Cluster Computing*, 28(7), 459. <https://doi.org/10.1007/s10586-025-05219-x>

- Sharma, P., Mirzan, S. R., Bhandari, A., Pimpley, A., Eswaran, A., Srinivasan, S., & Shao, L. (2020). Evaluating Tree Explanation Methods for Anomaly Reasoning : A Case Study of SHAP. 2.
- Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., Bagheri, M., & Djukic, P. (2022). Prior Knowledge based Advanced Persistent Threats Detection for IoT in a Realistic Benchmark. In GLOBECOM 2022 IEEE Global Communications Conference (pp. 3551–3556).
- Srikanth Yadav, M., & Kalpana, R. (2019). Data preprocessing for intrusion detection system using encoding and normalization approaches. Proceedings of the 11th International Conference on Advanced Computing, ICoAC 2019, 265–269.
- Wah, Y. B., Ibrahim, N., Hamid, H. A., Abdul-Rahman, S., & Fong, S. (2018). Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy. *Pertanika Journal of Science and Technology*, 26(1), 329–340.
- Xuan, C. Do, Duong, L. Van, & Nikolaevich, T. V. (2021). Detecting APT attacks based on network traffic using machine learning. *Journal of Web Engineering*, 20(1), 171–190.
- Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., Khan, R. T., Kaiser, M. S., & Mahmud, M. (2021). Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey. *IEEE Access*, 9, 94668–94690. <https://doi.org/10.1109/ACCESS.2021.3089681>
- Zhu, H., Wang, H., Lam, C. T., Hu, L., Ng, B. K., & Fang, K. (2024). Rapid APT Detection in Resource-Constrained IoT Devices Using Global Vision Federated Learning (GV-FL). *Communications in Computer and Information Science*, 1961 CCIS, 568–581.