**APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE**
e-ISSN: 2773-5141

**AITCS**

Vol. 7 No. 1 (2026) p. 1071-1088
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

# Advanced Persistent Threat (APT) Detection Based on Network Traffic Analysis using Machine Learning Approach

**Raqiebah Riza Faisal Riza[1], Isredza Rahmi A Hamid[1]***

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: rahmi@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.058

**Article Info**

Received: 3 August 2025
Accepted: 15 June 2026
Available online: 30 June 2026

**Keywords**

Advanced Persistent Threats (APTs), APT Detection, Network Traffic Analysis, Machine Learning, Random Forest Algorithm

**Abstract**

Advanced Persistent Threats (APTs) are stealthy and sophisticated cyberattacks targeting sensitive data and critical infrastructure. Traditional detection systems relying on signatures or anomaly-based methods face challenges in detecting evolving threats like APTs due to their stealthy, multi-stage behaviour and ability to mimic normal traffic patterns. These approaches often struggle to detect previously unseen attacks, adapt to new threat tactics, or maintain accuracy without generating high false positives. This research develops an APT detection model based on network traffic analysis using machine learning approach. The APT detection model leverages labelled datasets including Linux-APT-Dataset-2024 as Dataset A and APT Alerts as Dataset B. We analyse network traffic features such as Domain Name System (DNS) queries, traffic anomalies, and protocol deviations using Random Forest algorithm. We split into 3 sets of data based on Top 5, 10 and 15 features. The proposed APT detection model based on network traffic analysis achieved improved performance compared to existing work, with accuracy reaching 85.37% on Dataset A (Top 15 features) and 99.64% on Dataset B (Top 5 features). In addition, the model successfully reduced false positive rates, reaching as low as 44.33% on Dataset A and 0.34% on Dataset B. These findings highlight the model's robustness, offering cybersecurity professionals and organizations a reliable mechanism to detect and mitigate APT activities effectively.

## 1. Introduction

Advanced Persistent Threats (APTs) are defined as highly sophisticated cyberattacks that achieve sustained, covert access to target networks, primarily aimed at exfiltrating sensitive data or disrupting critical infrastructure. The severity of these threats is highlighted by Mandiant's M-Trends report, which revealed that in 2024, approximately 56% of successful compromises were first detected by external organizations. Moreover, the median dwell time for these adversaries, denoting the period of undetected presence within a network, extended to 11 days. These statistics emphasize the urgent requirement for the development of enhanced detection capabilities [1]. Several successful breaches of Critical Infrastructure (CI) have been attributed to Advanced Persistent Threats (APTs). Notable examples include Stuxnet, which targeted uranium enrichment facilities in Iran [2],[3]. Deep Panda is a sophisticated cyber-espionage campaign [4],[5]. Next is Epic Turla, a prominent APT operation [6],[7]. Others is the Oldsmar water treatment plant attack which endangered over 15,000 residents on

Florida's west coast by exposing them to potential water poisoning [8]. These attacks are complex and persistent that utilize advanced techniques to avoid detection over extended periods [9]. Detecting and mitigating APTs is crucial to safeguarding national security, preventing large-scale data breaches, and addressing the increasing sophistication of cyberattacks in an ever-evolving threat landscape.

Traditional methods to detect APT attacks such as signature-based and anomaly-based systems have limitations [10]. The signature-based methods fail against new or evolving threats [11], while anomaly-based detection systems often produce high false-positive rates [12]. Machine learning has emerged as a promising solution, analysing large datasets to identify unusual behaviours and provide more adaptive approach to threat detection [13]. This research proposed a machine learning-based approach using the Random Forest algorithm to analyse network traffic, enhancing detection accuracy and providing comprehensive view of APT activities [14].

This research aims to detect APT attacks based on abnormal network traffic behaviors, using features such as Flow Duration, Total Fwd Packet, Fwd Packet Length Mean, Flow Bytes/s, and Average Packet Size to capture variations in traffic patterns associated with multi-stage APT activities. We analyse domain and IP activity using the Linux-APT-Dataset 2024 which contains 6,000 samples, and the APT Alerts dataset with 4,000 samples. The feature selection method is done using Random Forest algorithm. We split into 3 sets of data based on Top 5, 10 and 15 features for both datasets. Then, the APT detection model is evaluated in terms of accuracy, precision, recall, false positives, and F1-Score using the Random Forest algorithm. This research will benefit cybersecurity professionals by addressing the limitations of existing detection systems. Moreover, this ensures better protection of sensitive data and critical infrastructure. It also provides an APT detection model that is adaptable to various cyber threats. The remainder of this paper is organized as follows; Section 2 discusses related work, Section 3 describes the methodology, Section 4 explains the result and discussion, and Section 5 explains the conclusion and future work.

## 2. Related Work

This section discusses about Advanced Persistent Threats (APTs), APT detection approach, APT features, related work in APT and comparison related work with the proposed work.

### 2.1 Advanced Persistent Threat (APT)

Advanced Persistent Threat (APT) is a critical and growing form of cyberattack characterized by their prolonged infiltration, persistence, and strategic focus on high-value assets [9]. Typically, APT is executed by nation-state actors or well-funded cybercriminal organizations. APTs aim to steal sensitive data, disrupt operations, or compromise critical infrastructure [9]. Unlike other cyberattacks, APTs employ advanced evasion techniques and follow a multi-stage lifecycle, including reconnaissance, initial compromise, lateral movement, and persistence [13]. The frequency and sophistication of APT attacks have increased significantly, with 38 of the 143 MITRE-tracked APT organizations active internationally in 2023, targeting critical sectors such as government, energy, healthcare, and banking [14]. Furthermore, more than 69% of firms have reported severe cyberattacks, including those related to APT activities. Notably, botnets, a key component in many APT campaigns, demonstrated lengthy endurance, with command-and-control communication lasting an average of 85 days [14]. Globally, thousands of compromises, particularly within government and healthcare, have been attributed to APT-related attacks, which commonly aim to exfiltrate financial data and intellectual property. [15][19]. Geopolitical conflicts such as the Russia-Ukraine war have further fuelled APT activity targeted critical infrastructure and disrupted essential services [16]. The urgent need for innovative detection strategies arises from APTs' inherent ability to mimic legitimate network traffic and continually evolve. This characteristic renders traditional signature-based methods ineffective against novel variants [15] and causes anomaly-based systems to generate unmanageably high false-positive rates [17]. This study focuses on APT detection to address the challenge of identifying stealthy, long-duration attacks that often evade traditional security systems. By leveraging advanced machine learning techniques, the research aims to improve early detection and response, thereby minimizing potential damage to organizational assets.

### 2.2 Advanced Persistent Threats (APT) Detection Approach

This section provides a comprehensive discussion of the various APT detection approaches, specifically focusing on signature-based and anomaly-based methods. Each approach is examined based on its key characteristics, strengths, and limitations. Understanding these detection mechanisms is essential for designing effective strategies to identify and mitigate APT activities within network environments.

#### 2.2.1 Signature Based

Signature-based approach works by comparing incoming network data or system activity against a database of predefined signatures, which represent characteristics of previously identified attacks. These signatures are derived from extensive analysis of historical attack patterns and typically include specific sequences of network packets, exploit payloads, or system event logs. Signature-based detection is useful for identifying well-documented threats quickly and efficiently with minimal false positives. However, its primary limitation lies in its inability to detect new, unknown, or zero-day attacks. It can only identify threats that included in its signature database [18]. This method often employed as the first layer of defence in hybrid systems, where it filters out obvious threats before more resource-intensive behaviour-based or anomaly detection systems are applied. Work by [19] applies the signature-based approach for Advanced Persistent Threat (APT) detection relies on recognizing known patterns of malicious behaviour. Work by [15] and [19] applies the signature-based approach for Advanced Persistent Threat (APT) detection relies on recognizing known patterns of malicious behaviour.

### 2.2.2 Anomaly Based

Contrasting signature-based methods which rely on predefined patterns of known attacks, anomaly-based detection approach uses heuristic or statistical analysis to monitor and classify system activity as normal or abnormal. This approach is particularly effective in detecting unknown or zero-day attacks because it identifies irregularities in behaviour rather than matching specific attack signatures. APTs often employ a "low and slow" strategy, blending their activity with legitimate network operations to avoid detection. By examining patterns over extended periods and correlating historical and real-time data from diverse sources, anomaly-based systems can detect subtle signs of APT activities. However, this approach is challenged by the high volume of data and the tendency to generate false positives.[20] To address these challenges, integrating big data analytics enables the processing of large and diverse datasets, enhancing the precision and reducing false positives, ultimately improving the detection of APTs [21]. Work by [17] and [22] presents the anomaly-based detection approach for Advanced Persistent Threats (APTs) focuses on identifying deviations from normal behaviour patterns within a system.

## 2.3 Advanced Persistent Threats (APT) Features

This section provides a comprehensive discussion of the various APT features categorized by their type, along with their key characteristics, strengths, and weaknesses. These features are crucial for identifying and analysing Advanced Persistent Threats within a network environment. Table 1 summarizes the details of Network-based, Domain-based, and IP-based APT features. While domain-based and IP-based features provide valuable, targeted insights into specific malicious indicators, network-based features outperform them for Advanced Persistent Threat (APT) detection due to their broad visibility and inherent ability to detect covert behaviors. As shown in Table 1, network-based analysis captures behavioral anomalies across all network layers [24], providing a more comprehensive understanding of activity than domain-based (which may struggle with dynamic or encrypted domains) and IP-based (vulnerable to IP rotation) analyses. Critically, network-based features are beneficial for detecting stealthy or low-volume threats [25], which is a key aspect of APTs designed for long-term and clandestine activities. Even though they require significant computational resources and may result in higher false-positive rates without advanced analysis [19, 25], network-based features' ability to discern complex, multi-stage behavioral patterns across the entire network traffic makes them indispensable for detecting the sophisticated and evolving nature of APTs.

**Table 1** *APT Features*

| Feature Type | Features | Strengths | Weaknesses |
|---|---|---|---|
| Network-based | • Traffic volume spikes [19].<br>• Protocol anomalies: DNS, HTTP abuse [19].<br>• Connection duration and timing patterns [19]. | • Captures behavioural anomalies across all network layers [24].<br>• Useful for identifying stealthy or low-volume threats [25]. | • High false-positive rates without advanced analysis [19].<br>• Demands real-time monitoring and robust computational resources [25]. |

**Table 1** *(cont.)*

| Feature Type | Features | Strengths | Weaknesses |
|---|---|---|---|
| Domain-based [26] | • DNS query: resolved IP count, distinct country count, silent IP ratio.<br>• Name server: server count, country count.<br>• Domain Generation Algorithms (DGAs). | • Effective in identifying suspicious domains used in C&C.<br>• Highlights anomalous server and DNS patterns.<br>• Identifies covert domains. | • May struggle with highly dynamic or encrypted domains.<br>• Requires integration with machine learning for accuracy. |
| IP-based [26] | • Outbound traffic volume and frequency.<br>• Unusual geographic locations or IP changes.<br>• Correlation of domain and IP behaviours. | • Identifies suspicious IP addresses and potential data exfiltration.<br>• Highlights unusual traffic patterns.<br>• Effective against IP rotation with temporal analysis. | • Resource-intensive for large-scale networks.<br>• Requires historical data for accurate behavioural analysis. |

## 2.4 Related Work in Advanced Persistent Threats (APT)

Work by [26] presents a robust approach to APTs through the analysis of network traffic using the Random Forest algorithm. The work utilized a dataset comprising 79,910 benign domains collected from established and reputable online services, and 84,167 malicious domains sourced from platforms such as PhishTank, Command-and-Control (C&C) domain blacklists, and other collaborative threat intelligence feeds. The features extracted included domain-based and IP-based attributes, focusing on abnormal behavior patterns to distinguish between benign and malicious activities. The model achieved an accuracy of 92.54%, a precision of 97.89%, recall of 88.31%, and an F1-score of 92.85%. Additionally, the model demonstrated a False Positive Rate (FPR) of 0.61%, a True Negative Rate (TNR) of 98.39%, and a False Negative Rate (FNR) of 11.69%, showcasing its effectiveness in detecting APT-related activities while minimizing false positives and negatives. This study highlights the potential of machine learning techniques in providing reliable and efficient solutions for APT detection.

Work by [27] provides a method for detecting suspicious Advanced Persistent Threat (APT) behaviors by thoroughly analyzing Domain Name System (DNS) operations. Recognizing that APTs frequently employ DNS for crucial functions including command-and-control (C2) communication, data exfiltration, and reconnaissance, their research focuses on detecting unusual patterns and variations in DNS queries and responses. This methodology emphasizes the importance of DNS traffic as a valuable source of indicators for detecting sophisticated, stealthy attacks that may circumvent typical perimeter defenses.

Work by [28] offered a new and extensive dataset for testing intrusion and Advanced Persistent Threat (APT) detection solutions in Linux settings. Titled "Advanced Persistent Threat (APT) and Intrusion Detection Evaluation Dataset for Linux Systems 2024," this resource addresses the important need for current, relevant data to train and test advanced detection models. The availability of such a dataset is crucial for academics and developers who want to build more robust and accurate solutions capable of detecting complex and developing threats.

Work by [29] different machine learning models were studied for their efficacy in detecting Advanced Persistent Threat (APT) attacks. Their study especially contrasted with the performance of a deep learning algorithm and a Bayesian Network model in detecting APT activities. The study's goal was to determine which methodology is better at accurately detecting malicious behaviors indicative of APTs within a system, thereby providing insights into effective algorithm selection for real-world deployment.

## 2.5 Comparison Related Work with the Proposed Work

Table 2 shows the comparison between various related works in Advanced Persistent Threat (APT) detection and the methodology proposed in this study, highlighting key similarities and differences across datasets, features, techniques, approaches, and achieved results.

**Table 2** *Comparison Related Work with the Proposed Work*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1071-1088

| Work | Dataset | Features | Technique | Approach | Result |
|---|---|---|---|---|---|
| A | Windows Event Log Datasets | Domain-based | Support vector machine | Anomaly-based detection | Recall: 100%, Precision: 100%, Accuracy: 100% |
| B | • 79910 benign domains (from the most well-known domain names on the Internet).<br>• 84167 phishing domains (from Phish Tank, C&C domains, Malicious domains list). | Network traffic, Domain-based and Ip-based | Random forest | Abnormal behaviour analysis | Accuracy: 92.54%, Precision: 97.89%, Recall: 88.31%, F1-Score: 92.85% , FPR: 0.61%, TNR: 98.39%, FNR: 11.69% |
| C | total 4,907,147,146 pieces of initial data of 47 days DNS request records of Jilin University Education Network | Domain-based | Graph-based anomaly detection | Graph-based approach | Accuracy: 97.6%, Recall: 96.8%, FPR: 2.3% |
| D | Linux-APT-Dataset-2024 | Domain-based, Network-based, IP-based | APT Simulations | anomaly detection, behavioural analysis, and signature-based detection | - |

A: Detecting APT attacks against Active Directory using Machine Leaning [27]

B: Detecting APT Attacks Based on Network Traffic Using Machine Learning [26]

C: Discovering Suspicious APT Behaviors by Analyzing DNS Activities [28]

D: Advanced Persistent Threat (APT) and intrusion detection evaluation dataset for linux systems 2024 [29]

## 3. Methodology

This section explains the methodology used to design Advanced Persistent Threat (APT) model. We provide an overview of APT detection model, the performance metric used, the hardware and software requirements.

### 3.1 Advanced Persistent Threats (APTs) Detection Model

The Advanced Persistent Threats (APTs) detection model is a structured framework designed to identify APT-related activities within network traffic. It comprises of six phases: Raw data, pre-processing, features extraction, feature selection, classification of algorithm and APT detection model. Each phase plays a critical role in ensuring accurate and reliable detection of malicious activities. Fig 1 shows the Advanced Persistent Threats (APTs) detection model.

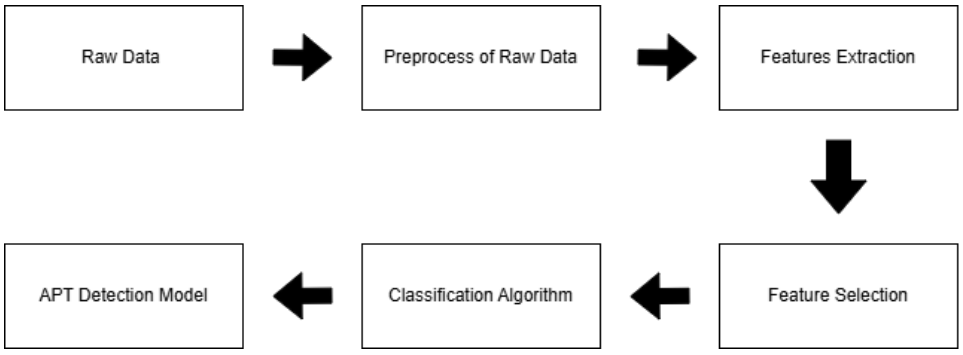


**Fig. 1** *Advanced Persistent Threats (APTs) Detection Model*

### 3.1.1 Raw Data

Raw data is composed from two publicly available datasets: the Linux-APT-Dataset-2024 [28] and the Dataset of APT Alerts from Loughborough University [29]. These datasets provide a mix of benign and APT-related network activities, ensuring a comprehensive training base for the detection model. Both datasets are structured and labelled to differentiate benign traffic from APT-related activities, making them suitable for supervised learning approaches. The combination of these datasets allows for a general approach to detecting APT behaviours in network traffic. Table 3 shows a summary of the datasets used in this study, detailing the distribution of benign and malicious samples.

**Table 3** *Dataset Summary*

| Dataset | Benign | Malicious | Total |
|---|---|---|---|
| Linux-APT-Dataset-2024 | 67,236 | 19,454 | 86,690 |
| APT Alerts dataset | 16,842 | 12,400 | 29,242 |

### 3.1.2 Preprocessing of Raw Data

Preprocessing is critical for improving the dataset's quality before training. It involves cleaning the data, handling missing values, normalizing numerical features, and encoding categorical ones to ensure compatibility with machine learning algorithms. The pre-processing data is used for feature extraction, where significant APT network-based features such as domain anomalies, traffic volume fluctuations, and unusual protocol patterns are identified.

### 3.1.3 Feature Extraction

In this phase, features were extracted from the raw network traffic to represent the behavior of each flow. These features capture statistical, header-based, and time-based patterns that help differentiate between normal and malicious (APT-related) traffic. The datasets were already structured in a flow format, so feature extraction was performed by processing each flow entry and calculating various metrics based on packet counts, byte totals, segment sizes, and inter-arrival times. To reduce dimensionality and retain only the most relevant features, Recursive Feature Elimination (RFE) was applied using a Random Forest classifier. RFE ranked the features by importance and recursively eliminated the least important ones. This process helped identify the top 15 most influential features for both datasets, which were later used for training and evaluation. A total of 84 features were extracted from both Dataset A and Dataset B, including attributes such as:

1. Flow Duration
2. Total Fwd Packet
3. Fwd Packet Length Max
4. Bwd Packet Length Mean
5. Flow Bytes/s
6. Packet Length Variance
7. Fwd Header Length
8. Bwd Init Win Bytes, and many others.

### 3.1.4 Feature Selection

The feature selection phase employs Recursive Feature Elimination (RFE) in combination with a Random Forest classifier to reduce the dimensionality of the dataset while preserving the most relevant features for APT detection. RFE works by recursively eliminating the least important features based on the model's feature importance scores, allowing the system to focus on attributes that contribute the most to prediction accuracy.

In this study, feature selection was conducted in three configurations: selecting the top 5, top 10, and top 15 most important features. These subsets were later used to evaluate model performance and determine the optimal balance between accuracy and computational efficiency. By reducing the number of features, this method helps prevent overfitting and improves the model's ability to generalize to new, unseen data.

### 3.1.5 Classification Algorithm

In this phase, the Random Forest algorithm was used as the primary classification method for detecting Advanced Persistent Threat (APT) activity. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the majority vote for classification. It is widely recognized for its high accuracy, robustness to overfitting, and effectiveness in handling large, high-dimensional datasets which making it well-suited for network traffic analysis.

The choice of Random Forest was based on several key factors. First, it efficiently handles both numerical and categorical data, which is important given the diverse nature of network traffic features. Second, it can rank feature importance, which was useful during the feature selection phase using Recursive Feature Elimination (RFE). Third, Random Forest is less prone to overfitting compared to single decision tree models, making it more suitable for real-world data where noise and imbalance may be present. Lastly, Random Forest has demonstrated strong performance in intrusion detection and anomaly detection applications, aligning well with the goal of this research.

To train and evaluate the model, the dataset was divided into two subsets: training data and testing data. The training data, which comprised 70% of the total dataset, was used to teach the model how to recognize patterns associated with APT and normal traffic. During this phase, the algorithm learned the relationships between input features and their corresponding class labels. The remaining 30% of the dataset was reserved as testing data. This subset was used to evaluate the model's ability to generalize and perform on previously unseen data, ensuring that the model does not merely memorize the training patterns but can accurately detect threats in real-world scenarios.

### 3.1.6 APT Detection Model

The Advanced Persistent Threat (APT) detection model in this research is developed using the Random Forest machine learning algorithm. Random Forest is an ensemble classifier that constructs multiple decision trees during training and combines their outputs through a majority voting process. This technique improves classification accuracy and reduces the risk of overfitting, making it well-suited for complex cybersecurity problems such as APT detection.

Each decision tree in the Random Forest is trained on a randomly selected subset of the data using the "bootstrap aggregation" (bagging) technique. Additionally, at each split in a tree, only a random subset of features is considered, which increases diversity among trees and contributes to the robustness of the model. Once trained, the Random Forest makes a prediction by aggregating the outputs of all decision trees which is choosing the class that receives the most votes.

Mathematically, if the Random Forest consists of T decision trees, each tree $\boldsymbol{h_t(x)}$ provides a classification result for an input instance $\boldsymbol{x}$. Equation 1 shows the final prediction $\boldsymbol{H(x)}$ where $\boldsymbol{h_t(x)}$ is the output of the $\boldsymbol{t^{th}}$ decision tree. The class with the highest number of votes among all trees is selected as the final prediction.

$$H(x) = majority\_vote\{h_1(x), h_2(x), \dots, h_T(x)\} \quad (1)$$

This ensemble approach enables the model to detect subtle and complex patterns in network traffic that may be indicative of APT behavior. By combining multiple trees, Random Forest provides both high accuracy and resilience to noisy or imbalanced data, which are common characteristics in real-world network environments.

## 3.2 Performance Metric

The model's performance is then assessed using various metrics, including accuracy, precision, recall, F1-score, and False Positive Rate (FPR). These metrics are derived from a confusion matrix, which helps identify the model's ability to correctly classify benign and malicious activities while minimizing false positives and negatives. Through these steps, the APT detection model is fine-tuned to reliably detect APT behaviours in real-world network environments.

### 3.2.1 Accuracy

Accuracy measures the overall correctness of the model by comparing the number of correct predictions (both benign and APT-related) to the total number of predictions, as shown in Equation 2. For APT detection, high accuracy indicates the model's ability to effectively distinguish between benign and malicious traffic. In this study, accuracy serves as a reliable performance metric, as the datasets used contain a balanced distribution of benign and APT traffic.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

### 3.2.2 Precision

Precision evaluates the proportion of true positives (correctly identified APT activities) to total positive predictions, as shown in Equation 3, highlighting the model's ability to minimize false positives. In APT detection, a high precision score indicates that the system generates fewer false alarms, enabling security analysts to focus on genuine threats rather than being overwhelmed by irrelevant alerts.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

### 3.2.3 Recall

Recall measures the model's ability to identify all actual APT-related activities, as shown in Equation 4, with high recall ensuring fewer undetected attacks. For APT detection, high recall is essential to ensure that the system captures as many APT-related activities as possible, reducing the likelihood of undetected attacks. A low recall score may leave organizations vulnerable to persistent threats.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

### 3.2.4 F1-Score

The F1-score shows in Equation 5 represents the harmonic mean of precision and recall. It provides a comprehensive measure of a model's performance by balancing both metrics. In the context of APT detection, a high F1-score indicates that the model can accurately identify malicious activities while minimizing false positives and false negatives, thus ensuring reliable threat detection.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

### 3.2.5 False Positive Rate (FPR)

The False Positive Rate (FPR) shows in Equation 6 measures the proportion of benign traffic that is incorrectly classified as APT-related. A low FPR is critical to avoid alert fatigue, a condition where security teams become overwhelmed by frequent false alarms and may begin to ignore or overlook real threats. Reducing FPR ensures that the alerts generated by the system are more accurate, enabling faster and more focused responses to genuine APT incidents.

$$FPR = \frac{FR}{FP + TN} \quad (6)$$

## 3.3 Hardware and Software Requirements

The hardware used for the Advanced Persistent Threat (APT) detection model consists of an HP Laptop 14s-cf2029TU, featuring an Intel Celeron N4020 CPU and 8GB of RAM. The laptop runs on Windows 11 Pro for Workstations. The software requires for developing the APT detection model includes Python for constructing the model, KNIME for data pre-processing tasks such as cleaning, normalizing, and encoding datasets, and SQLite for managing the second dataset. Additionally, the Scikit-learn library is employed for training and evaluating the detection model, utilizing machine learning tools like the Random Forest algorithm. This combination of hardware and software ensures a robust and efficient platform for implementing and validating the APT detection system.

# 4. Performance Analysis

This section discusses the performance analysis on APT detection proposed experiment. We provide an overview of experimental setup, feature extraction and selection using Recursive Feature Elimination (RFE), constructing feature matrix and result and discussion.

## 4.1 Experimental Setup

The experimental setup of the Advanced Persistent Threat (APT) detection model involves a structured process starting from raw data to model evaluation. This step-by-step pipeline ensures that the model is trained and tested on relevant and high-quality data, enhancing its effectiveness in detecting complex APT patterns. Two publicly available datasets were used for this research: the Linux-APT-Dataset-2024 (referred to as Dataset A) and the APT Alerts Dataset from Loughborough University (referred to as Dataset B). Dataset A is a synthetic dataset created by simulating APT behaviours in a Linux-based network environment. It captures multi-stage APT behaviours and includes a diverse range of malicious traffic. Dataset B consists of real-world APT alerts and provides operational realism, supporting the evaluation of the model's generalization capabilities. Table 3 summarizes the number of benign and malicious samples in each dataset.

Before applying machine learning algorithms, the raw data underwent a thorough preprocessing phase. This involved cleaning the dataset by removing duplicates and null values, normalizing numerical attributes, and

ensuring all feature formats were consistent. Since both datasets were already in a structured flow-based format, the focus was on ensuring compatibility with the feature extraction and model training processes.

Feature extraction involved deriving statistical, temporal, and header-based features from each flow record. A total of 84 features were extracted from both Dataset A and Dataset B, including key attributes such as Flow Duration, Total Fwd Packet, Fwd Packet Length Max, Bwd Packet Length Mean, Flow Bytes/s, Packet Length Variance, Fwd Header Length, and Bwd Init Win Bytes. These features were selected based on their relevance in capturing traffic behaviours typical of APT activity.

To reduce dimensionality and improve model efficiency, Recursive Feature Elimination (RFE) was applied in combination with a Random Forest classifier. RFE works by recursively removing the least important features based on the model's feature importance scores. This method allowed the model to focus on the most informative attributes while improving generalization and reducing overfitting. Feature selection was conducted in three configurations: top 5, top 10, and top 15 features, which were later used to train and compare model performance.

The Random Forest algorithm was used as the classification method for this APT detection model due to its robustness, accuracy, and ability to handle high-dimensional data. The pre-processed datasets were split into 70% training data and 30% testing data. The training data was used to train the model to identify the relationship between network behaviours and their corresponding labels (benign or APT). The testing data, which was not seen during training, was used to assess the model's ability to generalize to new, unseen traffic flows. Table 4 shows the summary of the dataset analysis conducted prior to model development, while Table 5 provides an overview of the dataset usage in terms of training and testing distribution.

After training, the model was evaluated using key performance metrics, including Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). These metrics were calculated for each of the feature configurations (top 5, 10, and 15 features) and across both datasets. The results demonstrated that the top 15 features provided the best performance overall, especially in balancing high accuracy with low FPR. The evaluation confirmed the model's effectiveness in detecting APT threats with a strong balance of sensitivity and precision.

**Table 4** *Summary of datasets analysis*

| Dataset | Training Set | | Testing Set | |
|---|---|---|---|---|
| | Malicious | Normal | Malicious | Normal |
| A (Linux-APT-Dataset-2024) | 13,628 | 47,055 | 5,826 | 20,181 |
| B (APT Alerts dataset_) | 8,680 | 11,789 | 3,720 | 5,053 |

The experimental setup for the Advanced Persistent Threat (APT) detection model was established using a combination of tools and libraries essential for data analysis, model development, and evaluation. Scikit-learn was used to implement the machine learning algorithm (Random Forest), while Pandas and NumPy were utilized for data manipulation and handling. Matplotlib and Seaborn were employed for visualizing feature distributions and model performance results. For pre-processing raw data, KNIME provided a graphical interface to perform cleaning, normalization, and transformation tasks efficiently. Additionally, SQLite was used for managing intermediate datasets in a lightweight database environment, and Joblib facilitated model serialization and persistence for reuse and deployment. The proposed detection model applies machine learning techniques to analyze network traffic and identify APT-related threats. This approach addresses the limitations of traditional detection systems such as signature-based and anomaly-based methods, which often struggle to detect stealthy, previously unseen APT attacks due to their evolving and multi-stage nature.

**Table 5** *Summary of datasets used*

| Dataset | Total Samples | Training Samples (70%) | Testing Samples (30%) |
|---|---|---|---|
| Linux-APT-Dataset-2024 | 74,390 | 48,383 | 26,007 |
| APT Alerts dataset | 29,242 | 20,469 | 8,773 |
| Total | 103,632 | 68,852 | 34,780 |

## 4.2 Feature Extraction and Selection Using Recursive Feature Elimination (RFE)

To identify the most relevant network features for detecting Advanced Persistent Threat (APT) activities, Recursive Feature Elimination (RFE) was applied to both Dataset A and Dataset B. RFE is a supervised feature selection technique that works by recursively training a model and eliminating the least important features at each iteration based on the model's feature importance. In this study, a Random Forest classifier was used within RFE due to its robustness and built-in feature importance ranking. This iterative process continues until the

desired number of features is retained, ensuring the selection of only the most informative attributes for model training.

RFE operates by fitting a model repeatedly and ranking features using the importance score $\boldsymbol{W}$, where features with the lowest contribution are removed at each step. Equation 7 shows the summary of RFE process where $\boldsymbol{W_j}$ denotes the importance score of the j-th feature.

$$RFE\ Importance\ Rank = argmin_j |W_j| \tag{7}$$

Using this technique, the top 15 feature were selected based on their contribution to classification performance. These include both standard flow-based features and newly proposed features specific to this research, such as Src Port, Dst Port, Fwd Header Length, Bwd Init Win Bytes, Average Packet Size, and Bwd Packet Length Min. These newly proposed features were not emphasized in existing literature (Papers A, B, and C) but proved highly valuable for APT detection. Figure 2 shows top features selected by RFE for Dataset A while Figure 3 shows top features selected by RFE for Dataset B.

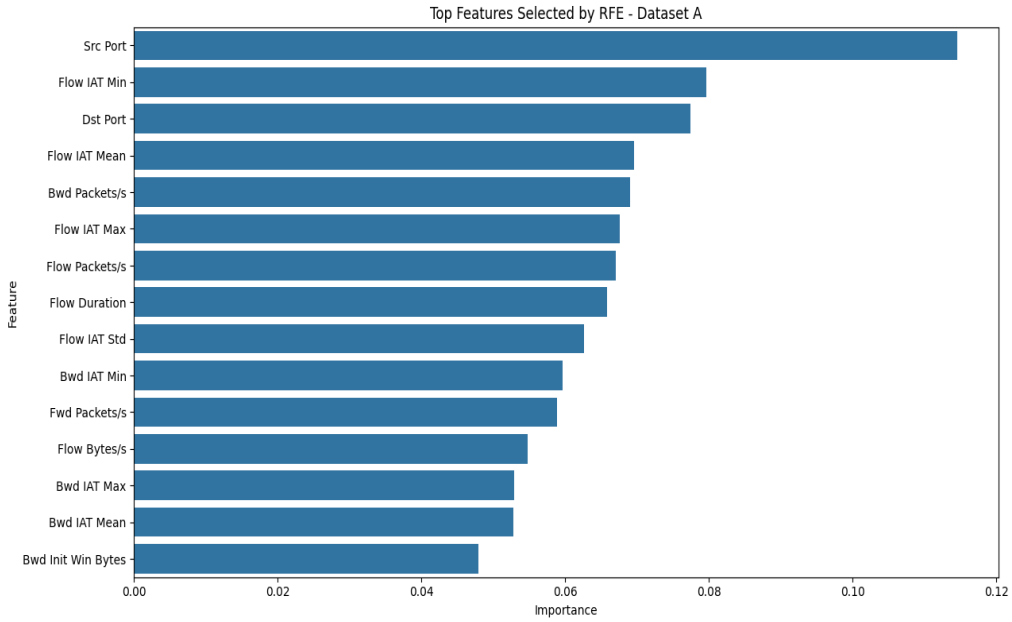


**Fig. 2** *Figure Top Features Selected by RFE (Dataset A)*

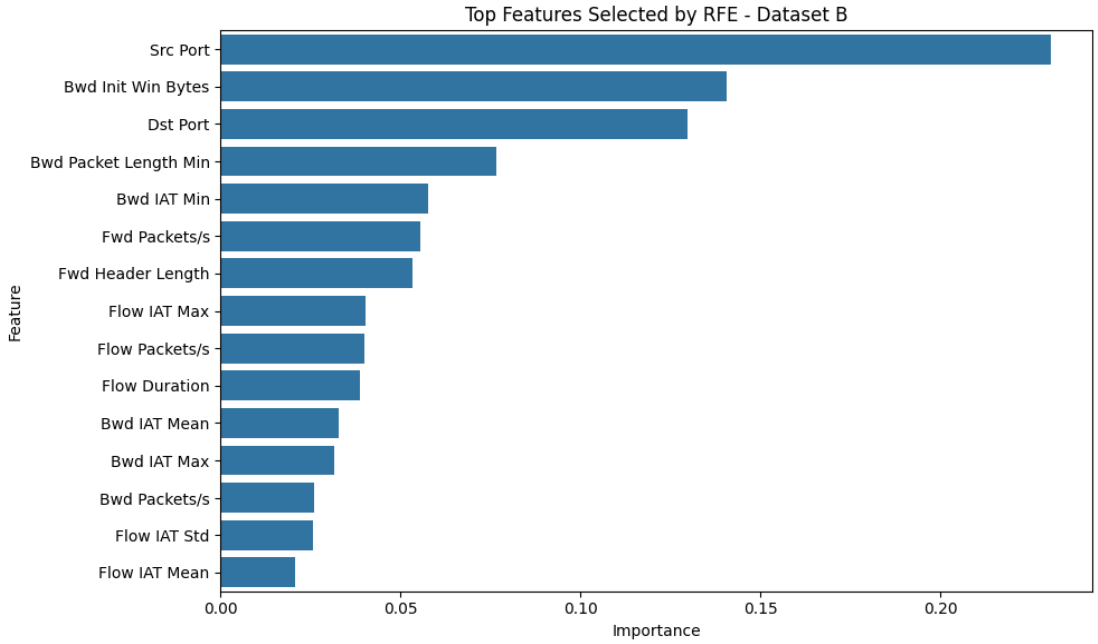


**Fig. 3** *Figure Top Features Selected by RFE (Dataset B)*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1071-1088

### 4.2.1 APT Network Based Features

The Advanced Persistent Threat (APT) detection model incorporates a carefully selected set of features derived from network traffic analysis. These features were identified using Recursive Feature Elimination (RFE), an effective feature selection method that ranks input variables based on their importance in contributing to the model's performance. This approach ensures the inclusion of only the most relevant features, reducing redundancy and enhancing model efficiency.

The Linux-APT-Dataset-2024 (Dataset A) and APT Alerts dataset (Dataset B) are categorized into three sets based Top 5, Top 10, and Top 15 features as shown in Table 7 to Table 12. We prepare 12 sets of data as displays in Table x.

**Table 6** *Sets of Data*

| | Feature Set | Dataset A | Dataset B |
|---|---|---|---|
| Work A | Top 5 Proposed Work A | / | / |
| | Top 10 Proposed Work A | / | / |
| | Top 15 Proposed Work A | / | / |
| Work B | Top 5 Proposed Work B | / | / |
| | Top 10 Proposed Work B | / | / |
| | Top 15 Proposed Work B | / | / |
| Work C | Top 5 Proposed Work C | / | / |
| | Top 10 Proposed Work C | / | / |
| | Top 15 Proposed Work C | / | / |
| APT Detection Model | Top 5 Proposed Work Features | / | / |
| | Top 10 Proposed Work Features | / | / |
| | Top 15 Proposed Work Features | / | / |

**Table 7** *Comparison Top 5 Features (Dataset A)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Flow IAT Min |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Flow IAT Mean |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd Packets/s |

**Table 8** *Comparison Top 10 Features (Dataset A)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Flow IAT Min |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Flow IAT Mean |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd Packets/s |
| 6. | Fwd Packet Length Max | Fwd Packet Length Mean | Flow IAT Mean | Flow IAT Max |
| 7. | Fwd Packet Length Min | Bwd Packet Length Mean | Flow IAT Std | Flow Packets/s |
| 8. | Fwd Packet Length Mean | Flow Bytes/s | Flow IAT Max | Flow Duration |
| 9. | Fwd Packet Length Std | Flow Packets/s | Flow IAT Min | Flow IAT Std |
| 10. | Bwd Packet Length Max | Packet Length Variance | Fwd IAT Mean | Bwd IAT Min |

**Table 9** *Comparison Top 15 Features (Dataset A)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Flow IAT Min |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Flow IAT Mean |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd Packets/s |
| 6. | Fwd Packet Length Max | Fwd Packet Length Mean | Flow IAT Mean | Flow IAT Max |
| 7. | Fwd Packet Length Min | Bwd Packet Length Mean | Flow IAT Std | Flow Packets/s |
| 8. | Fwd Packet Length Mean | Flow Bytes/s | Flow IAT Max | Flow Duration |
| 9. | Fwd Packet Length Std | Flow Packets/s | Flow IAT Min | Flow IAT Std |
| 10. | Bwd Packet Length Max | Packet Length Variance | Fwd IAT Mean | Bwd IAT Min |
| 11. | Bwd Packet Length Min | Fwd Header Length | Bwd IAT Mean | Fwd Packets/s |
| 12. | Bwd Packet Length Mean | Bwd Header Length | Packet Length Mean | Flow Bytes/s |
| 13. | Bwd Packet Length Std | - | Packet Length Std | Bwd IAT Max |
| 14. | Flow Bytes/s | - | Fwd Packets/s | Bwd IAT Mean |
| 15. | Flow Packets/s | - | Bwd Packets/s | Bwd Init Win Bytes |

**Table 10** *Comparison Top 5 Features (Dataset B)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Bwd Init Win Bytes |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Bwd Packet Length Min |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd IAT Min |

**Table 11** *Comparison Top 10 Features (Dataset B)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Bwd Init Win Bytes |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Bwd Packet Length Min |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd IAT Min |
| 6. | Fwd Packet Length Max | Fwd Packet Length Mean | Flow IAT Mean | Fwd Packets/s |
| 7. | Fwd Packet Length Min | Bwd Packet Length Mean | Flow IAT Std | Fwd Header Length |
| 8. | Fwd Packet Length Mean | Flow Bytes/s | Flow IAT Max | Fwd IAT Max |
| 9. | Fwd Packet Length Std | Flow Packets/s | Flow IAT Min | Flow Packets/s |
| 10. | Bwd Packet Length Max | Packet Length Variance | Fwd IAT Mean | Flow Duration |

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 1071-1088

**Table 12** *Comparison Top 15 Features (Dataset B)*

| No. | Paper A | Paper B | Paper C | Proposed Work |
|---|---|---|---|---|
| 1. | Flow Duration | Flow Duration | Flow Duration | Src Port |
| 2. | Total Fwd Packet | Total Fwd Packet | Fwd Packet Length Max | Bwd Init Win Bytes |
| 3. | Total Bwd Packet | Total Bwd Packet | Fwd Packet Length Mean | Dst Port |
| 4. | Total Length of Fwd Packet | Total Length of Fwd Packet | Bwd Packet Length Max | Bwd Packet Length Min |
| 5. | Total Length of Bwd Packet | Total Length of Bwd Packet | Bwd Packet Length Mean | Bwd IAT Min |
| 6. | Fwd Packet Length Max | Fwd Packet Length Mean | Flow IAT Mean | Fwd Packets/s |
| 7. | Fwd Packet Length Min | Bwd Packet Length Mean | Flow IAT Std | Fwd Header Length |
| 8. | Fwd Packet Length Mean | Flow Bytes/s | Flow IAT Max | Fwd IAT Max |
| 9. | Fwd Packet Length Std | Flow Packets/s | Flow IAT Min | Flow Packets/s |
| 10. | Bwd Packet Length Max | Packet Length Variance | Fwd IAT Mean | Flow Duration |
| 11. | Bwd Packet Length Min | Fwd Header Length | Bwd IAT Mean | Bwd IAT Mean |
| 12. | Bwd Packet Length Mean | Bwd Header Length | Packet Length Mean | Bwd IAT Max |
| 13. | Bwd Packet Length Std | - | Packet Length Std | Bwd Packets/s |
| 14. | Flow Bytes/s | - | Fwd Packets/s | Flow IAT Std |
| 15. | Flow Packets/s | - | Bwd Packets/s | Flow IAT Mean |

Among the features identified by RFE, several newly proposed network flow-based characteristics played a crucial role in enhancing the model's ability to discern malicious APT traffic from legitimate network activities. Specifically, Source Port (Src Port) and Destination Port (Dst Port), while seemingly basic, offer significant insights into the nature of communication. APTs often utilize unusual or non-standard ports for command and control (C2) communications or data exfiltration, making the analysis of these ports vital for anomaly detection. Deviations from typical port usage patterns can serve as strong indicators of malicious activity.

Furthermore, Flow Inter-Arrival Time Minimum (Flow IAT Min) and Flow Inter-Arrival Time Mean (Flow IAT Mean) provide critical temporal insights into the flow of network packets. APT attacks, particularly those involving stealthy data exfiltration or C2 beaconing, often exhibit distinct timing patterns that differ significantly from normal network traffic. A consistently small Flow IAT Min might indicate a rapid burst of communication, potentially for data transfer, while changes in Flow IAT Mean could signify covert channels or slow-and-low exfiltration attempts. By capturing these subtle temporal characteristics, the model can effectively identify communication irregularities indicative of APT presence.

The consistent appearance of these newly proposed features alongside established metrics across both datasets, as illustrated in Tables 13-18, reflects their strong predictive value in distinguishing between normal and malicious APT-related traffic. The use of RFE not only helped streamline the feature set but also emphasized domain-relevant traffic behaviours, making the resulting detection model more interpretable and efficient for practical deployment. This detailed feature analysis reinforces the robustness and targeted nature of our proposed APT detection approach.

**Table 13** *Network Traffic Features Used*

| Features | Description | Value |
|---|---|---|
| Source IP Address | The IP address of the sender in the network traffic. | IPv4/IPv6 format (e.g., 192.168.1.1) |
| Destination IP Address | The IP address of the recipient in the network traffic. | IPv4/IPv6 format (e.g., 192.168.1.2) |
| Traffic Volume | The total amount of data transferred in a session. | Integer (e.g., 1024 bytes, 1 MB) |
| Packet Size | The size of each packet transmitted in the network. | Integer (e.g., 64 bytes, 1500 bytes) |
| Anomaly Score | A numerical score indicating potential anomalies in the traffic pattern. | Float (e.g., 0.3, 0.85) |

Table 13 shows an overview of the key network traffic features used in this study to support APT detection. These features are crucial for analysing and distinguishing between normal and potentially malicious network behaviour. The Source IP Address and Destination IP Address indicate the origin and target of each network session, respectively, and are typically formatted in IPv4 or IPv6 (e.g., 192.168.1.1). Traffic Volume captures the total amount of data transmitted during a session, which helps identify abnormal data transfers that may signal exfiltration attempts. Packet Size refers to the size of individual packets transmitted over the network. Variations in packet size can be indicative of evasion or tunnelling techniques often associated with APT behaviour. Lastly, the Anomaly Score represents a numerical value that reflects the likelihood of abnormal activity in each session, often based on deviations from learned traffic baselines. Collectively, these features provide a foundational representation of network behaviour that aids in the machine learning-based detection of Advanced Persistent Threats.

## 4.3 Constructing Feature Matrix

The Random Forest algorithm works by constructing multiple decision trees, each trained on a randomly selected subset of the dataset, to analyse and classify features indicative of Advanced Persistent Threats (APTs). Each tree in the forest makes an individual prediction, and the final output is determined by a majority vote (for classification) or averaging (for regression). This ensemble method helps reduce overfitting, improves generalization, and captures complex, non-linear relationships between features.

For classification tasks, Random Forest commonly uses Gini Index and Entropy to determine the best feature to split a node in a decision tree. For example, suppose a dataset contains 60% malicious traffic and 40% benign. Equation 8 shows the Gini index calculated.

$$Gini = 1 - (0.6^2 + 0.4^2) = 0.48 \tag{8}$$

A lower Gini value indicates a purer node. Similarly, Entropy measures the disorder or uncertainty in the data. Equation 9 shows entropy is calculated using the same example.

$$Entropy = \sum_{i=1}^{C} - P_i * log_2(P_i) \tag{9}$$

Both metrics help the algorithm choose the feature that best splits the data into distinct classes. The one with the lowest Gini or Entropy after the split is usually selected.

For regression tasks (though not the focus of this study), Random Forest uses Mean Squared Error (MSE) to evaluate splits. MSE measures the average squared difference between predicted and actual values. For instance, Equation 10 shows MSE value if the actual APT severity scores are [2, 4, 3] and the model predicts [2.5, 3.5, 3].

$$MSE = \frac{(2.5 - 2)^2 + (3.5 - 4)^2 + (3 - 3)^2}{3} = 0.17 \tag{10}$$

In conclusion, Random Forest combines these splitting criteria to build multiple decision trees, enabling more accurate and stable detection of APT activities. By capturing intricate feature interactions, it helps reduce overfitting and improves both robustness and overall model performance.

## 4.4 Result and Analysis

This section presents the evaluation metrics and results of the Random Forest-based APT detection model. Five performance metrics were applied to assess the model's performance: Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). Each model was tested using the top 5, 10, and 15 most important features, which were ranked by feature importance for both Dataset A and Dataset B. The results are divided into three parts which are Evaluation Metric Results, Comparison with Existing Work, and Proposed Features.

### 4.4.1 Evaluation Metrics Result

The Random Forest-based APT detection model was evaluated using five performance metrics: Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). Each model was tested using the top 5, 10, and 15 most important features which ranked by feature importance for both Dataset A and Dataset B.

Each set was independently evaluated for its impact on model performance across five metrics: accuracy, precision, recall, F1-score, and false positive rate (FPR). The results clearly indicate that the top 15 features

produced the most optimal balance between precision and recall, delivering robust performance even in complex threat environments.

**Table 14** *Research Accuracy*

| Dataset | Feature Set | | |
|---|---|---|---|
| | Top 5 | Top 10 | Top 15 |
| A (Linux-APT-Dataset-2024) | 0.8354 | 0.8464 | 0.8537 |
| B (APT Alerts dataset_) | 0.9964 | 0.9954 | 0.9910 |

**Table 15** *Research Precision*

| Dataset | Feature Set | | |
|---|---|---|---|
| | Top 5 | Top 10 | Top 15 |
| A (Linux-APT-Dataset-2024) | 0.8673 | 0.8757 | 0.8785 |
| B (APT Alerts dataset_) | 0.9955 | 0.9940 | 0.9862 |

**Table 16** *Research Recall*

| Dataset | Feature Set | | |
|---|---|---|---|
| | Top 5 | Top 10 | Top 15 |
| A (Linux-APT-Dataset-2024) | 0.9294 | 0.9338 | 0.9408 |
| B (APT Alerts dataset_) | 0.9962 | 0.9955 | 0.9932 |

**Table 17** *Research F1-Score*

| Dataset | Feature Set | | |
|---|---|---|---|
| | Top 5 | Top 10 | Top 15 |
| A (Linux-APT-Dataset-2024) | 0.8972 | 0.9038 | 0.9086 |
| B (APT Alerts dataset_) | 0.9959 | 0.9947 | 0.9897 |

**Table 18** *Research False Positive Rate (FPR)*

| Dataset | Feature Set | | |
|---|---|---|---|
| | Top 5 | Top 10 | Top 15 |
| A (Linux-APT-Dataset-2024) | 0.4847 | 0.4516 | 0.4433 |
| B (APT Alerts dataset_) | 0.0034 | 0.0046 | 0.0106 |

### 4.4.2 Comparison Analysis

This section presents a comparative analysis between the proposed APT detection model and three existing research works referred to as Paper A, Paper B, and Paper C. The comparison focuses on model performance when using features identified in those studies versus the top-ranked features selected in this research from Dataset A and Dataset B. Each paper's feature set was re-implemented and tested under identical conditions using the Random Forest classifier, enabling a fair and consistent evaluation. Performance was assessed across five metrics: Accuracy, Precision, Recall, F1-score, and False Positive Rate (FPR).

As presented in Table 19 and Table 20, the proposed top 15 features achieved the best overall performance when compared to the feature sets from Paper A, Paper B, and Paper C across both datasets. Specifically, for Dataset A, the top 15 proposed features recorded the highest accuracy of 85.37%, precision of 87.85%, recall of 94.08%, and F1-score of 90.86%, along with the lowest false positive rate (FPR) of 44.33%. Similarly, for Dataset B, the top 15 proposed features demonstrated exceptional results, with an accuracy of 99.10%, precision of 98.62%, recall of 99.32%, F1-score of 98.97%, and a low FPR of 1.06%. These results indicate that the proposed features not only enhance classification accuracy but also significantly reduce false positives, which is critical in operational environments. This validates the effectiveness of the proposed feature selection strategy and confirms its advantage over existing works. Furthermore, the consistent high performance across multiple metrics supports the successful achievement of all three research objectives: identifying relevant APT features from network traffic,

designing an effective detection model using machine learning (Random Forest), and evaluating the model using comprehensive performance metrics.

**Table 19** *Performance Comparison on Proposed Work (Dataset A)*

| Feature Set | Accuracy | Precision | Recall | F1-Score | False Positive Rate (FPR) |
|---|---|---|---|---|---|
| Paper A Top 5 Features | 0.7057 | 0.6546 | 0.5620 | 0.6952 | 0.0793 |
| Paper A Top 10 Features | 0.7027 | 0.6466 | 0.5564 | 0.6946 | 0.0800 |
| Paper A Top 15 Features | 0.7009 | 0.6372 | 0.5377 | 0.6919 | 0.0865 |
| Paper B Top 5 Features | 0.7173 | 0.6639 | 0.5746 | 0.7083 | 0.0757 |
| Paper B Top 10 Features | 0.7167 | 0.6475 | 0.5510 | 0.7066 | 0.0779 |
| Paper B Top 15 Features | 0.7150 | 0.6198 | 0.5496 | 0.7030 | 0.0893 |
| Paper C Top 5 Features | 0.7390 | 0.7025 | 0.5916 | 0.7278 | 0.0715 |
| Paper C Top 10 Features | 0.7269 | 0.7019 | 0.5867 | 0.7265 | 0.0722 |
| Paper C Top 15 Features | 0.7110 | 0.7000 | 0.5698 | 0.7144 | 0.0745 |
| Top 5 Proposed Work Features | 0.8354 | 0.8673 | 0.9294 | 0.8972 | 0.4847 |
| Top 10 Proposed Work Features | 0.8464 | 0.8757 | 0.9338 | 0.9038 | 0.4516 |
| Top 15 Proposed Work Features | 0.8537 | 0.8785 | 0.9408 | 0.9086 | 0.4433 |

**Table 20** *Performance Comparison on Proposed Work (Dataset B)*

| Feature Set | Accuracy | Precision | Recall | F1-Score | False Positive Rate (FPR) |
|---|---|---|---|---|---|
| Paper A Top 5 Features | 0.8696 | 0.8033 | 0.7830 | 0.8699 | 0.0830 |
| Paper A Top 10 Features | 0.8506 | 0.8027 | 0.7500 | 0.8650 | 0.0955 |
| Paper A Top 15 Features | 0.8390 | 0.7958 | 0.7201 | 0.8204 | 0.0100 |
| Paper B Top 5 Features | 0.8801 | 0.8024 | 0.7870 | 0.8802 | 0.0770 |
| Paper B Top 10 Features | 0.8755 | 0.8115 | 0.7888 | 0.8900 | 0.0799 |
| Paper B Top 15 Features | 0.8580 | 0.8250 | 0.7902 | 0.8999 | 0.0889 |
| Paper C Top 5 Features | 0.8811 | 0.8172 | 0.8034 | 0.8952 | 0.0753 |
| Paper C Top 10 Features | 0.8777 | 0.8227 | 0.8123 | 0.8958 | 0.0780 |
| Paper C Top 15 Features | 0.8548 | 0.8435 | 0.8398 | 0.8980 | 0.0793 |
| Top 5 Proposed Work Features | 0.9964 | 0.9955 | 0.9962 | 0.9959 | 0.0034 |
| Top 10 Proposed Work Features | 0.9954 | 0.9940 | 0.9955 | 0.9947 | 0.0046 |
| Top 15 Proposed Work Features | 0.9910 | 0.9862 | 0.9932 | 0.9897 | 0.0106 |

A: A Study on the Importance of Features in Detecting Advanced Persistent Threats Using Machine Learning [27]
B: Enhancing Network Threat Detection with Random Forest-Based Intrusion Detection Systems [28]
C: Cyber Attack Detection thanks to Machine Learning Algorithms [29]

## 5. Conclusion

The detection of Advanced Persistent Threats (APTs) remains a critical challenge in modern cybersecurity, as their sophistication and persistence often bypass traditional detection methods. This research aimed to develop an efficient APT detection model based on network traffic analysis using the Random Forest algorithm. By leveraging comprehensive datasets, specifically the Linux-APT-Dataset-2024 and the APT Alerts dataset, the proposed approach focuses on identifying anomalies in network behaviour to enhance detection accuracy while minimizing false positives.

The findings of this study demonstrate the potential of machine learning techniques in addressing key limitations of existing methods, including their inability to detect zero-day threats and high false-positive rates. The proposed model's emphasis on analysing network-based, domain-based, and IP-based features offers a more holistic approach to identifying APT activities. Through the rigorous application of evaluation metrics such as Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR), and the strategic use of Recursive Feature Elimination (RFE) for feature selection, our model consistently achieved high detection performance across both datasets. Notably, the top 15 features selected by RFE demonstrated superior and balanced performance

compared to feature sets from existing works, as evidenced by the comparative analysis in Tables 19 and 20. This highlights the effectiveness of our feature engineering and selection strategy in identifying the most relevant indicators of APT activity.

This research contributes significantly to the field of cybersecurity by providing an adaptable and scalable solution for detecting APTs in real-time. The robust performance observed underscores the efficacy of combining network traffic analysis with the Random Forest algorithm for proactive threat detection.

Future work will involve further refining the model to improve its detection capabilities, exploring additional datasets for diverse threat scenarios, and incorporating advanced ensemble techniques to enhance performance metrics. With the ever-evolving nature of cyber threats, this study underscores the importance of continuous innovation in APT detection methods to safeguard critical infrastructure and sensitive information.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

## Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

*The authors confirm contribution to the paper as follows:* ***study conception and design:*** *R. R. Faisal Riza, I. R. A. Hamid;* ***data collection:*** *R. R. Faisal Riza, I. R. A. Hamid;* ***analysis and interpretation of results:*** *R. R. Faisal Riza, I. R. A. Hamid;* ***draft manuscript preparation:*** *R. R. Faisal Riza, I. R. A. Hamid. All authors reviewed the results and approved the final version of the manuscript.*

## References

[1] Che Mat, N. I., Jamil, N., Yusoff, Y., & Mat Kiah, M. L. (2024). A systematic literature review on advanced persistent threat behaviors and its detection strategy. Journal of Cybersecurity, 10(1), tyad023.

[2] S. Kriaa, M. Bouissou and L. Pièvre-Cambacédès, "Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments,"

[3] Alperovitch, D. (2011). Revealed: operation shady RAT (Vol. 3, p. 2011). McAfee.

[4] Noor, U., Anwar, Z., Amjad, T., & Choo, K. R. (2019). A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise. Future Generation Computer Systems, 96, 227–242.

[5] MITRE ATT&CK (2021). https://attack.mitre.org/groups/G0009/

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

[7] Kaspersky, (2021). The Epic Turla (snake/Uroburos) attacks. https://www.kaspersky.co.uk/resource-center/threats/epic-turla-snake-malware-attacks

[8] Kaufman, E. K., Adeoye, S., & Batarseh, F. A. (2023). Leadership for CyberBioSecurity: The Case of Oldsmar Water.

[9] Fahad, M., Airf, H., Kumar, A., & Hussain, H. K. (2023). Securing Against APTs: Advancements in Detection and Mitigation. BIN: Bulletin Of Informatics, 1(2)

[10] Hallaji, E., Razavi-Far, R., & Saif, M. (2025). A Study on the Importance of Features in Detecting Advanced Persistent Threats Using Machine Learning. arXiv preprint arXiv:2502.07207.

[11] Manandhar, P. (2014). A practical approach to anomaly-based intrusion detection system by outlier mining in network traffic. Masdar Institute of Science and Technology.

[12] Amaru, Y., Wudali, P., Elovici, Y., & Shabtai, A. (2024). RAPID: Robust APT Detection and Investigation Using Context-Aware Deep Learning. arXiv preprint arXiv:2406.05362.

[13] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), 160.

[14] Mohsin, M., & Ali, A. (2021). A comparative analysis of machine learning algorithms for network intrusion detection. International Journal of Information Security, 20(2), 153-166.

[15] Cho, D. X., & Nam, H. H. (2019). A method of monitoring and detecting APT attacks based on unknown domains. Procedia Computer Science, 150, 316-323.

[16] Sommer, P., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. 2010 IEEE European Symposium on Security and Privacy, 1-16.
[17] Kwon, S. J., & Kim, H. J. (2019). A machine learning approach for detecting advanced persistent threats in network traffic. Future Generation Computer Systems, 100, 407-418.
[18] Delplace, A., Hermoso, S., & Anandita, K. (2020). Cyber attack detection thanks to machine learning algorithms. arXiv preprint arXiv:2001.06309.
[19] AL-Aamri, A. S., Abdulghafor, R., Turaev, S., Al-Shaikhli, I., Zeki, A., & Talib, S. (2023). Machine Learning for APT Detection. Sustainability, 15(18), 13820.
[20] Kwon, H. -Y., Kim, T., & Lee, M. -K. (2022). Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods. Electronics, 11(6), 867. https://doi.org/10.3390/electronics11060867
[21] Ezefosie, N., & Ohemu, M. F. (2021). A data-driven anomaly-based behavior detection method for advanced persistent threats (APT). International Journal of Science and Research (IJSR), 10(8), 663-667.
[22] Alazab, M., & Venkatraman, S. (2019). Detecting Advanced Persistent Threats using Network Traffic Analysis. Journal of Network and Computer Applications, 133, 1-13.
[23] Moustafa, N., & Slay, J. (2016). The Evaluation of Network Traffic Anomaly Detection Techniques for Advanced Persistent Threats. Proceedings of the 2016 IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 1-6.
[24] Do Xuan, C., Dao, M. H., & Nguyen, H. D. (2020). APT attack detection based on flow network analysis techniques using deep learning. Journal of Intelligent & Fuzzy Systems, 39(3), 4785-4801.
[25] Matsuda, W., Fujimoto, M., & Mitsunaga, T. (2018, November). Detecting apt attacks against active directory using machine leaning. In 2018 IEEE Conference on Application, Information and Network Security (AINS) (pp. 60-65). IEEE.
[26] Yan, G., Li, Q., Guo, D., & Meng, X. (2020). Discovering suspicious APT behaviors by analyzing DNS activities. Sensors, 20(3), 731.
[27] Karim, S. S., Afzal, M., Iqbal, W., & Al Abri, D. (2024). Advanced Persistent Threat (APT) and intrusion detection evaluation dataset for linux systems 2024. Data in Brief, 54, 110290.
[28] Ooi, H. N., & Ab Rahman, N. H. (2021). A Comparative Study between Deep Learning Algorithm and Bayesian Network on Advanced Persistent Threat(APT) Attack Detection. Applied Information Technology And Computer Science, 2(2), 219-235. https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2324
[29] Abdelaziz, M. T., Radwan, A., Mamdouh, H., Saad, A. S., Abuzaid, A. S., AbdElhakeem, A. A., ... & Darweesh, M. S. (2025). Enhancing network threat detection with random forest-based NIDS and permutation feature importance. Journal of Network and Systems Management, 33(1), 2.