Advanced Persistent Threat Attack Detection using Clustering Algorithms

Ahmed Alsanad[1], Sara Altuwaijri[2]
Department of Information Systems
College of Computer and Information Sciences
King Saud University, Riyadh 11543, Saudi Arabia

***Abstract*—Advanced Persistent Threat (APT) attack has become one of the most complex attacks. It targets sensitive information. Many cybersecurity systems have been developed to detect the APT attack from network data traffic and request. However, they still need to be improved to identify this attack effectively due to its complexity and slow move. It gets access to the organizations either from an active directory or by gaining remote access, or even by targeting the Domain Name Server (DNS). Nowadays, many machine learning (ML) techniques have been implemented to detect APT attack by using the tools in the market. However, still, there are some limitations in terms of accuracy, efficiency, and effectiveness, especially the lack of labeled data to train ML methods. This paper proposes a framework to detect APT attacks using the most applicable clustering algorithms, such as the APRIORI, K-means, and Hunt's algorithm. To evaluate and compare the performance of the proposed framework, several experiments are conducted on a public dataset. The experimental results showed that the Support Vector Machine with Radial Basis Function (SVM-RBF) achieves the highest accuracy rate, reaching about 99.2%. This accurate result confirms the effectiveness of the developed framework for detecting attacks from network data traffic.**

***Keywords*—APT Attack detection; DNS; network; cybersecurity; clustering algorithms**

I. Introduction

People and organizations worldwide use technology for most of their daily activities. This change is called digital transformation, which requires organizations to profoundly transform their business model, infrastructure, processes, and culture. So, the usage of the Internet is increased [1]. Although technologies and the Internet make life easier, they have been used for harmful purposes. Cybersecurity crimes impact society [2] since these crimes occur through modern communication devices using internet connections. The actors who cause a cybercrime are called attackers, and they are different kinds and have multiple goals; one of those kinds is APT Attacks. APT stands for Advanced Persistent Threat [3], and it is one of the top cybersecurity concerns in enterprise networks [4]. APT means: Advanced, which means the attacker is stealing, targeting, and data-focused attacks [5]. Persistent means an attacker identifies the target to breach, hide, and exploit them [6]. Word Threat in APT means the extraction of critical data [5]. APT are complex, and they are well-planned security attacks [7]. So, its consequences will impact the organizations by stealing intellectual property, compromising and stealing sensitive information, stealing classified data, critical organizational infrastructures, and accessing diplomatic communication channels. Also, the ability to detect APT activity at the network level is heavily dependent on leveraging threat intelligence [8]. Attackers use multiple techniques to hide and infect the targets; the method is not limited to phishing, zero-day attack, waterhole attacks [3], and denial of service (DoS) attacks [9]. APT attack functions are developed to avoid detection as long as possible [10]. So, many techniques have been used to detect change controlling, sandboxing, and network traffic analysis [11].

Increasing the frequency of security breaches and cyberattacks on the Internet of Things (IoT) requires dependable security solutions [12]. In addition to firewalls, the Network Intrusion Detection System (NIDS) is the second network infrastructure security system that detects malicious activity and prevents attacks [13-15]. Moreover, security administrators typically choose password protection systems, encryption techniques, and access controls to protect the network. These measures, however, are insufficient to protect the system [16]. As a result, the administrators prefer to utilize Intrusion Detection Systems (IDSs) to monitor network traffic and detect malicious attacks [17-21]. For example, in [22], the authors proposed an over-sampling Principal Component Analysis (PCA) to address the anomaly detection problem.

Today, alert correlation is done using Security Information and Event Management (SIEM) systems such as Splunk, LogRhythm, and IBM QRadar [23]. They collect multiple log events and alert various sources. But the APT Attack has evolved to bypass security mechanisms that are difficult for technologies to find [24]. This paper studies how to detect APT attacks according to the framework. Currently, there is a significant potential for cyber-attack these days. A cyber-attack is intentionally exploiting computer systems, infrastructures, and networks. Cyber-attack has been done throw the attackers; the attackers are multiple kinds and category. These attackers are different from each other in terms of the goals and methods they use. Common types of cybersecurity attacks are malware attacks, Denial-of-service attacks, password attacks, and APT attacks. APT is a complex and multi-stage attack. Since its complex, they need many stages to meet their target by collecting information as much as possible and carefully [25]. Afterward, they will use their technique to reach what they need, such as phishing. Attackers then collect confidential data using multiple malware after they breach the network. Also, they use various techniques to send the data taken to another server.

Based on the NIST framework, cybersecurity programs have five primary functions. These functions are, identify, protect, detect, respond and recover. Some attackers will be known in the preserve or prevent phase, and others in the detection phase. In this paper, we focus on detecting APT attacks. That detecting APT attacks is challenging because it defeats and supersedes the premier defense devices by injecting their techniques as part of large normal traffic [25]. They are also closely linked to each other and are hidden, so it is usually too late to detect them. The attacker needs more time to efficiently distribute the attacker’s activities and behaviors, with a challenging possibility to be detected. So, for APT attack detection, we use multiple techniques and tools to detect it using an attack signature, monitoring the network, and collecting network information.

Several techniques are implemented to detect the APT attack by using the tools in the market. These techniques are either using artificial intelligence (AI) or machine learning (ML) methods. However, still, there are limitations in terms of accuracy, efficiency, and effectiveness, especially the lack of labeled data to train ML methods. Significantly, the APT attacks are brutal to be detected. They usually target sensitive and critical data. An organization infected and exploited by APT attacks will harm and lose many essential assets or data. APT attacks are very complex due to their lifecycle and evolution complexity.

This paper's scope is to develop a framework and apply it as a tool to identify and detect APT attacks. Using machine learning for analyzing the attacker's behavior, the framework is implemented to help the cybersecurity specialist, especially those working in the Security operation center (SOC), to know and detect if their organization is breached and hacked by APT attackers. This framework will minimize the harm and impact that the APT attack will do. Also, it will be more accessible to cybersecurity vendors to build their detection tools. Through the proposed framework, the research contributions to the field can be summarized as follows.

- A framework to detect APT attacks is proposed to tackle the lack of labeled data using unsupervised clustering algorithms such as the APRIORI algorithm, Hunt’s algorithm, and the K-means algorithm.

- The proposed framework is implemented on the CSE-CIC-IDS2018 dataset for achieving the performance of supervised learning of the ML models.

- A comparative study of the five ML classifiers is performed to detect the APT attacks.

- The framework's performance results are evaluated using several evaluation measures on the dataset.

The rest of the paper is structured as follows: Section II gives a background for the study. Section III presents the literature review of the previous work on detecting APT attacks. Section IV explains the proposed framework to identify and detect an APT attack. Section V introduces the experiments with findings and discussions. Finally, Section VI summarizes the conclusions and future research work.

II. Background

An APT attack's lifecycle is more complex than other kinds of attacks. A successful APT attack can be divided into multiple stages [26]. In the first stage, the attacker will define the target by determining who he will target and why he wants to target him. Next, he will select the team members and identify the required skills. Then the attacker will find the existing tools or develop new ones he/she needs. After that, the attacker will discover who has access to what he needs and what HW/SW will use. Then, the attacker will test if he/she can detect or not by deploying a miniature version of the tool, piloting a connectivity and alarm trail, check and spotting any weaknesses. Later on, he/she will launch full fledge attach on the victim’s platform.

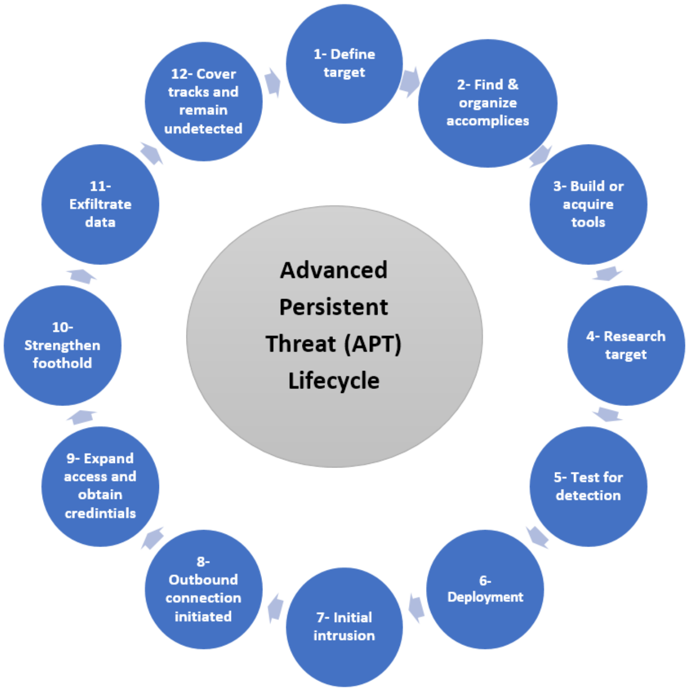
1- Define target
2- Find & organize accomplices
3- Build or acquire tools
4- Research target
5- Test for detection
6- Deployment
7- Initial intrusion
8- Outbound connection initiated
9- Expand access and obtain credintials
10- Strengthen foothold
11- Exfiltrate data
12- Cover tracks and remain undetected
Advanced Persistent Threat (APT) Lifecycle

Fig. 1. APT Lifecycle.

The first entrance will be shown in the network where the target is. After that, he will establish a secure connection from victim’s platform to his Command-and-Control Center. He will obtain credentials by creating a hidden Trojan on the victim platform. Then, he will start navigating through the rest of the platform to create more Trojans. After that, and once he gets what he was looking for, he will cover the tracks to remain undetected and make sure to clean up after himself. Fig. 1 shows and summarizes the stages of the APT attacker’s lifecycle.

III. Literature Review

This section discusses the previous work related to detecting APT attacks. An intrusion detection system (IDS) is an inevitable line of defense against cyber threats [27]. The challenge here is that IDS lacks typical evaluation methodologies to detect this attack. This section will do a literature review for detecting APT attacks and what the target is.

To detect APT attacks, two main approaches are commonly used, which are:

- Detection based on signature: It is a well-known technique based on the attack's signature [28] and low efficiency.
- Detection based on behavior: it is an enhanced technique that focuses on the attacker's signature and behavior [28], and its result is high efficiency and high processing costs.

The literature review of APT attack detection will be divided based on what the attacker targets the system, network, or domains.

A. *APT Attack Targeting Active Directory*

As Active Directory (AD) is a system that manages the organization's accounts for windows systems. This fact makes it a target for APT attacks since it is expected that domain admin privileges have been accessed by APT attacks.

To detect this attack, various research talks about it how to use machine learning and focus on the attacker’s signatures and characteristics [29], and they are:

- Detection using authentication files: Using machine learning (Unsupervised) to analyze authentication files or monitor abnormal user behavior.
- Process-based detection: utilizing backlist in conjunction with signature-based detection in the log files [29], then false negatives will arise case the attackers manipulate file names of the tools because the file name is the main element on the signature algorithm.
- Detection through network traffic monitoring: an example of the methods is Golden Ticket [29] by traffic monitoring. But this feature is not implemented for windows systems.

The proposed approach was for outlier detection using Domain Controllers logs with machine learning related to processes. The advantages of this kind of method are their ability to detect AD attacks with high accuracy by abusing the command and tools that attackers are using. Also, because it uses only Domain Controller logs, it is very cost-effective. The target is detecting attacks that require control of admin privileges of the Domain Administrator.

The algorithms used are machine learning utilizing existing data without any programming effort. With this unsupervised learning, there is no need to provide correct answers [29]. So, no need to analyze the attacker's behavior.

After evaluating the methods, machine learning was the most appropriate algorithm for their way. The other one was preprocessing for machine learning which describes the necessary preprocessing for machine learning. Any logs that show a particular feature, such as logs with blank values, need to be eliminated because they can be identified with no value. When the attackers disguise their identity as an official Domain Administrator account, and the hijacked Domain Administrator account uses tools or commands, the attacker also uses false detection. This means that Administrators use commands which are rarely used [29]. The APT Attack against AD is brutal to be detected since that attackers usually take advantage of processes and legitimate accounts [29].

B. *APT Attack and Intrusion Detection Event*

The prediction model for intrusion detection is based on events that show the probability of threat intrusion detection events through the prediction task [30]. After the analysis, it detects the attacks before or after a particular attack exists in a correlation [30]. By extracting the events of intrusions, a specific scenario is configured. When it takes place after detecting it, the next attack in the plan can be predicted by investigating at which stage of the attack scenario the intrusion detection events occur. That will result in enabling the prediction of the last threat [30].

The intrusion detection event based on the prediction model collects and pretreats intrusion detection events [30], extracts sessions and threads, creates scenarios of the attackers through correlation analysis, predicts intrusions, & expresses the analyzed results [30].

The prediction based on intrusion detection events leads to a search of an event on a scenario of the attacker when an intrusion detection event is detected [30]. When a single event occurs, other events can take place afterward. The issues that face intrusion detection events can be given as follows:

- Time required in prediction and verification of intrusion detection events: the daily average count of intrusion detection events was tremendous and incomparable with the duration of the collected data. So, it is necessary to extract successful attack events by time unit, attack type, and organization, distinguishing them from all intrusion detection events [30].
- Validity of prediction due to narrow gap in intrusion detection events: the time difference in the collected intrusion detection events verified their correlation was primarily within several seconds. Intrusion detection events in government organizations are managed by the enterprise system to monitor the database every five minutes [30]. So, the response to the events is primarily impossible, and the use of anticipated events is less.
- Intrusion detection data and intrusion detection rule: these rules are frequently added, modified, and deleted [30]. Even though those rules are changed, the sequential rules must be learned, and the rules must be applied to an independent prediction model based on intrusion detection events [30]. To do this task, a full-time employee needs to monitor and track it and be dedicated to this.
- Stability of intrusion detection system: the rule-based system used for monitoring cannot provide stability to detect the continuously changing types of attacks [30].

Based on the intrusion detection event model, prediction and verification of the events problems are not only of the time required, but the issues of cybersecurity threat prediction, such as problems in intrusion detection rules, intrusion detection data [30], and the stability of IDS systems. In addition, the main problem is that it requires automated monitoring detection to predict different APT attacks.

C. APT Attacks Targeting Network Infrastructure

Network infrastructure APT attacks are many. The first, called Moonlight Maze, targeted government networks [5]. The other one is called operation Aurora targeting cloud computers [5]. To detect those, the author wrote that many challenges would be faced; these challenges are [5]:

- Unsupervised anomaly-based detection approaches to discover all anomalies.
- Supervised alert correlation-based approaches decide whether some attacks are related or are a portion of a more advanced APT attack.

Evaluation and training of those approaches would require entire labeled traces of networks with widespread abnormal and intrusions behaviors [5] and need to be labeled specially for APT-correlated alerts. Also, there are several constraints in detecting an APT attack that targets network, such as:

- No one unique path in which all APT attack activities can be detected.
- Over time the APT attacks tend and adapt to use new tools and vulnerabilities.

The approaches of the IDS, including APT detection methods, consider the feature construction and selection stages as the first-time consuming step. The features can be built using machine learning and data mining methods or manually, such as association mining, sequence analysis, and frequent episode mining [5]. Some of their features categories are:

- Basic features: the essential attributes and features are collected from the connection of TCP/IP.
- Traffic features: the attributes and features that can be computed or extracted from concerning a window interval.
- Content features: the attributes and features that can be extracted from the data payload for suspicious behaviors.

The result of targeting network infrastructure should be focused on automated methods for APT attack detection. It can cover two types of use cases according to the essential infrastructure. In the first use case, the large enterprise networks are considered to have known attacks, such as GhostNet [5], Moonlight Maze, and attacks on cloud computing-based systems like Aurora Operation. Usually, the detection is based on the attack model. For the second use case, the goal network is typically used to extract sensitive information [5]. One of the achievements of this paper is the investigation and description of the existing methodologies and the detailed overview of APT detection approaches related to their infrastructure.

D. APT Attack to Get Remote Access

The APT attack will get remote access to the target by embedding malware, installing them on the target's device, connecting to the control server, and maintaining the control channel [31]. To maintain control of the contact, the heartbeat mechanism is also used [31]. They use HTTP, email protocol, and FTP [31] to get remote access. These protocols are standard protocols of application transport for communication to communicate between the inject sides and controls as hidden as possible to avoid security equipment inspection and audit [31]. Remote access is a perfect way for the anomaly to hide in the regular traffic since there are no variances between the communication of remote control and regular network application communication [32].

E. APT Attacks based on Domain Name Server

One of the techniques to detect APT attacks is analyzing the domain name server (DNS). This is because DNS request constitutes only a tiny fraction of the overall traffic of the network, making it appropriate for analysis and investigation the large-scale networks [33]. Also, DNS traffic contains many significant features to recognize domain names that might be associated with malicious events. These features can be more enriched with related information [25].

The DNS feature extraction can be used to achieve an effective detection of APT attacks. There are three kinds of these features host, time, and domain. The APT Unsupervised Learning Detection (AULD) framework is proposed to detect APT attacks [25] using the DNS features. It can detect suspicious DNS domains with APT attacks based on unsupervised machine learning. The first step is to preprocess the collected DNS request; ten features have been extracted based on host, time, and domain. AULD framework can analyze many DNS log files and obtain the list of APT attacks. Also, it can extract the host, time, and domain features from the DNS log data regarding the behaviors of attackers during an APT attack detection [25].

The results have shown that the framework could detect APT activities effectively [25]. The list of suspicious domains can be detected by cybersecurity experts to define the entire APT attack detection process [25]. Also, the AULD framework can enable cybersecurity experts to analyze suspicious domains and block APT events as soon as possible [25].

F. APT Attacks based on Accessing Unknown Domains

This section describes an architecture for detecting and monitoring APT attacks depending on access to unknown domains [28]. The architecture module of the APT attack detection and monitoring solution is shown in Fig. 3, and its methods are described as follows:

- Accuracy: APT attacks are prepared through email spam, social phishing, and email phishing [28] to reach their targets. The APT attack detection by unknown domains has a high accuracy result if one unknown domain has been detected. Others will send an alarm to users [28]. Accordingly, admins will take the appropriate action.
- Detection Time: APT attack detection can be handled in a real-time manner [28]. It is a very critical factor for preventing APT attacks at the early stages.

The algorithms for detecting an APT attack that targets these kinds are proposed as follows:

- Algorithms for detecting unknown domains: these algorithms will identify the malicious domains that are possibly suspicious domains of APT attacks [28].
- Algorithms for monitoring access to unknown domains: these algorithms will monitor suspicious activities. The unknown domains will be detected using this algorithm that will be checked by using the generation rule algorithms and simply monitoring techniques [28].

The system model used to monitor and detect the APT attacks of unknown domains is shown in Fig. 2. The model has a number of components, given as follows:

- Datacenter: the data center stores data, including weblogs, network traffic, and normalized data [28]. It gives information for monitoring and tracking network attacks. This extracted information is related to the activities and behaviors of the attackers.
- APT attacks monitoring and detection component: these components monitor and detect APT attacks using DNS logs [28]. The data center provides input for this component. This component includes the following:
  - Database: it is used to provide and store data, which is associated with the signatures of the attackers.
  - Processing component: This component implements the algorithms, methods, and techniques which are used for processing to detect APT attacks [28]. The output of this component is a set of APT attacks and suspicious domains.
- Alarm component: It is responsible for issuing warnings and alarms at different levels with evidence that the APT attacks penetrate the systems being monitored [28].

The architecture module of the APT attack detection and monitoring solution consists of the following components:

- Database: this includes:
  - Signatures of APT attacks database: it stores the signatures of all APT attacks.
  - Detecting result database: it saves the domains that are analyzed and collected by the unknown in the database [28].
  - Monitoring result database: the domains used for analysis DNS logs of unknown domains are stored in the database.

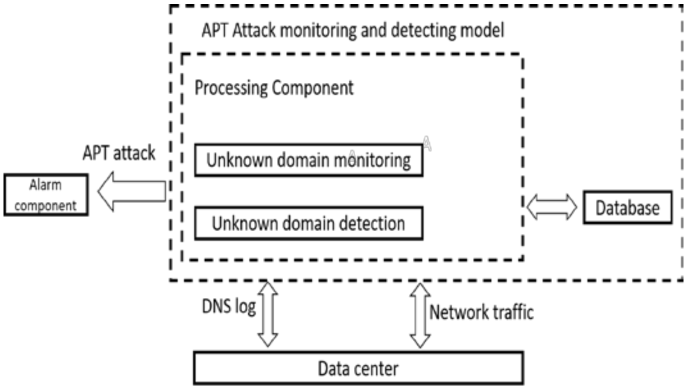

Fig. 2. Detection of APT Attacks from Unknown Domains.

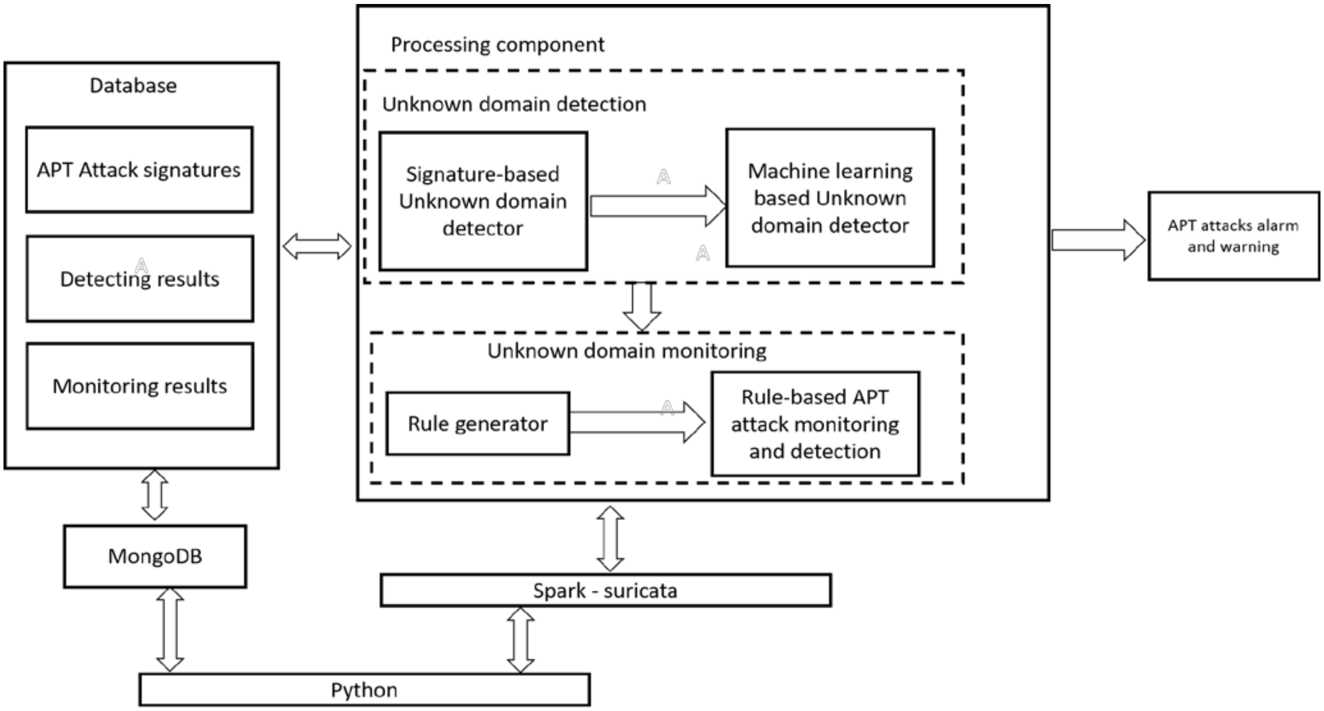
Processing component
Database
Unknown domain detection
APT Attack signatures
Signature-based Unknown domain detector
Machine learning based Unknown domain detector
APT attacks alarm and warning
Detecting results
Unknown domain monitoring
Monitoring results
Rule generator
Rule-based APT attack monitoring and detection
MongoDB
Spark - suricata
Python

Fig. 3. Architecture Module of the APT Attack Detection and Monitoring Solution [28].

- Unknown domain detection: it consist of a set of algorithms to detect and monitor unknown domains, and it includes:
  - Signature-based unknown domain detector: they collect and extract the signatures from the described APT attacks DNS logs. They are used as evidence for APT attack detection. It compares the domains signatures in the DNS logs with the collected actual APT attack signatures. If the signatures are matched, then these domains are malicious; otherwise, they are benign [28].
  - Machine learning-based unknown domain detector: this is done to identify unknown APT attack domains. A set of suspicious domains is provided as input, and a set of unknown malicious/benign domains is returned [28]. In this study, a clustering technique was employed.

As a result, the APT attack has multiple stages and steps of its implementation. If one stage fails, the whole APT will fail [28]. The method presented is for APT attacks that uses monitoring access to the unknown domains in a real-time manner in high efficiency and effectiveness.

The persistent nature of this kind of attack reveals the necessity of having precise analysis to measure the damages in the absence of proper diagnosis and treatment. This raises several concerns:

1) Continues activities from adversaries to breach victims' platforms and to seek the weakest link. This requires continuous monitoring activities and applying the rights update to the media.

2) It is not easy to detect the breaches once the advisories gain access to the victims' platform. This requires specialized tools and skilled human resources.

3) Recovery will take time to clean up all the resources because of the methods used during the breach.

4) Cost again this type of attack is high since it requires advanced detection and protection tools and continuous monitoring.

5) Skilled resource availability will be playing a significant role, and it has to be appropriately addressed.

IV. Proposed Framework to Identify and Detect an APT Attack

APT attacks are complex and hard to be detected. This paper introduces a framework for identifying and detecting APT attacks. The framework is an automated unsupervised machine learning [25], and the output is a set of suspicious DNS domains by analyzing the DNS features. This framework can report the suspicious domains to the security engineer and help the defenders detect faster APT attacks [25]. The framework is divided into four stages: the data collection stage, data preprocessing stage, feature extraction stage, and clustering stage. Fig. 4 shows the flowchart of the proposed framework.

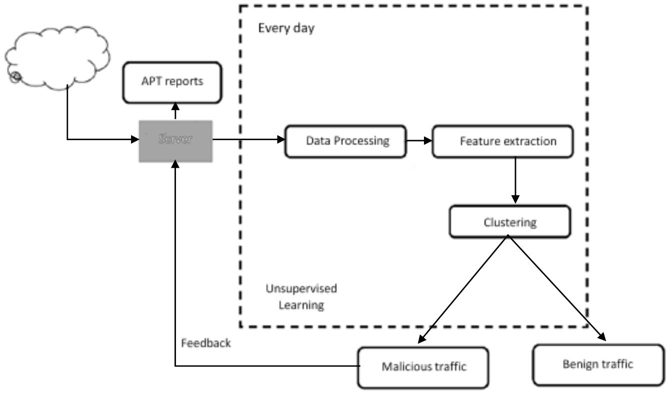

Fig. 4. Flowchart Diagram of the Proposed Framework.

The first part of this framework is data collection, which will collect DNS data log records for a certain period. When finding a precise time sequence for an IP of the internal host, the accessing date, the accessing domain, and other fields among the APT attacker's reports and giving some malicious domains, this will star detect the APT attack.

The second part is data processing; in this part, we will do the following [25]:

- By extracting a valid field and changing the format of the data in the data raw.
- Folding domain into the next level of domain.
- Deleting the whitelist of sites.
- Deleting famous websites within the internal network to get the experimental data.

A feature extraction will then be done by knowing the number of devices that get access to the domains, the domain's popularity, access time, automatic connection, domain age, and similarity of a domain. This is all based on the three types of features, which are time, host, and domain. The last part is the clustering process, and this is done according to the proper algorithm upon testing them such as K-mean clustering algorithm, or Hierarchical clustering, or Density-based clustering algorithms. The framework contains the following steps:

- Data preparation: This is the stage of data preprocessing in which unnecessary features and duplicate instances are removed in preparation for identification. Convert categorical attributes to numerical values through data digitization. Normalization for modifying the scale, type, and probability distribution of variables in a dataset is an example of a data transformation.
- Feature selection and reduction: using the PCA technique to pick the most relevant features subset approaches the detection phase as input.

Detection: On the CSE-CIC-IDS2018 dataset, we improved classification accuracy by utilizing KNN, decision tree, and two kernels (linear, RBF) with SVM machine learning classifiers, as well as a random forest classifier.

A. Model Evaluation

After the model is trained using the training data samples, it can pass into the test step. Inspecting how the model works in practical circumstances is the aim of testing. This stage allows us to evaluate the model's precision. In this study, the model attempts to identify the APT attack using the knowledge gained during the training step. The evaluation process is vital because it enables us to determine whether the model accomplishes the objective of classifying the network traffic. The previous procedures must be repeated until the requisite accuracy is attained if the model does not function as anticipated during the testing phase. As previously indicated, it should not use the same data that was used during the training phase. It needs to utilize a different data splitter from the data set for analysis.

The accuracy of the outcome is one of the classification measures taken into consideration for evaluating the trained models. When producing classification outputs, there are four possible outcomes: true positives, true negatives, false positives, and false negatives. These four outcomes are represented on a confusion matrix. The matrix can be created based on the results after classifying the test inputs, and each output can be classified as one of the potential outcomes. The model's accuracy is measured by the proportion of correct classification from the test data. The number of correctly classified instances divided by the total number of instances gives the result of accuracy. Additionally, classification models are assessed using additional metrics such as precision and recall.

B. Adopted Algorithms in the Framework

This section proposes an intrusion detection system based on machine learning algorithms. A Principal Component Analysis (PCA) algorithm is used for feature reduction. This method improves the performance detection task [22]. Traditionally, PCA reduces the feature dimension by linearly transforming original n-dimensional features into n orthogonal axis, as shown in Fig. 5. By projecting an observation onto each of these axes, a new set of n uncorrelated variables is created. The new feature vector is composed of a subset of these variables with a high eigenvalue. However, each derived feature requires n × n multiplications and the use of all original features to compute. The computation time for feature extraction will rise as a result of this. In this study, a PCA removes some unnecessary features from the feature set. The information extracted from the coefficients of the Principal Components (PC) is used for feature ranking and reduction.

The covariance matrix (*C*), of the *n*-dimensional features vector taken from positive training samples, is created first. The Principal Components are then determined using C's eigenvalues and eigenvectors (PCs). There are a total of n potential PCs. Each of the PCs has *n* coefficients, each of which is associated with a correlated feature from the original feature pool. The characteristic associated with the PC's largest coefficient is placed in the highest rank by starting with the first PC. The same technique is used on succeeding PCs to generate a list of features in descending order. A varying number of low-ranked features are deleted depending on the ranking to generate a subset of reduced features.

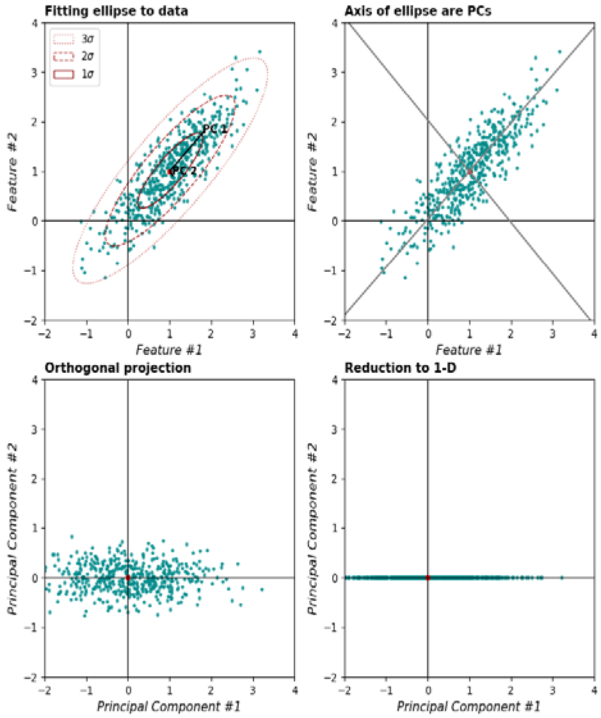

Fig. 5. PCA Feature Reduction by Linearly Transforming Original n-dimensional Features into n Orthogonal axis.

- Experiments have been conducted to determine the smallest number of characteristics that can accurately represent the entire feature set. After that, we achieved a comparative study of the five proposed classifiers, which are:
- Decision Tree (DT).
- Random Forest (RF).
- K-Nearest Neighbor (KNN).
- Support Vector Machine with Linear Function (SVM-Linear).
- Support Vector Machine with Radial Basis Function (SVM-RBF).

V. Experiments and Discussions

A. CSE-CIC-IDS2018 Dataset

In this section, we describe the CSE-CIC-IDS2018 dataset [1] used to evaluate the proposed framework. It includes detail on intrusions as well as protocol specifics. The Canadian Institute for Cybersecurity released its most recent dataset in 2018-2019. This dataset contains seven different forms of assaults: Botnet, infiltration, DoS, Heartbleed, DDoS, Brute force, and Web attacks. The compromised firms had 30 servers and 420 PCs, while the attacking infrastructure had 50 terminals.

The CICFlowMeter-V3 dataset [26] is collected traffic of AWS network and machine log files with more than 70 extracted features. The best way to test and evaluate the system framework is represented by the network's applications and the lowest level entities; it also refers to the move from static data to dynamic data, which is real-time traffic on the Amazon platform (AWS). Furthermore, the dataset was improved by taking into account the standards that were designed to produce CIC-IDS2017. In addition to the basic criteria, it has the following advantages:

- There are very few duplicate data records.
- Uncertain data is almost non-existent.
- The dataset is in CSV format so that it can be used immediately without further processing.

B. Evaluation Metrics

Some evaluation metrics such as confusion matrix, accuracy, detection rate, precision, recall, and F1-score are used to evaluate the effectiveness of the framework’s ML algorithms.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall\ (True\ Positive\ Rate\ (TPR)) = \frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-}Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

- Confusion matrix: In intrusion detection, a confusion matrix is a useful tool for predicting the type of network attack. It contains where TP refers to true positive instances (TP), true negative instances (TN), false positive instances (FP), and false negative instances (FN).
- Accuracy: The percentage of positive data cases detected correctly.
- Precision: The number of attacks correctly returned.
- Recall or True Positive Rate (TPR): The number of attacks the system returns.
- F1-score: In our approach, the rate of precision and recall:

C. Experimental Results

The CSE-CIC-IDS2018 dataset [1] is first preprocessed by eliminating eleven non-essential features such as the timestamp, average number of bulk rates, and number of times the PSH flag was set in packets. The parameters are set by default in all of the implemented algorithms in this report, with the exception of KNN, which uses the n nearest neighbor’s property (*n* = 3). The number of classes in the suggested algorithm was determined to be one (zero for non-attack types and one for attack types). The most recent dataset available was used for training and testing the CSE-CIC-IDS2018 dataset. In the trials, training and test data were divided into 80 percent and 20 percent to assess the performance results related to training and testing.

Although PCA aims to maximize the distance between data points, it has no concept of classes. The default libraries in Python programming language like the Scikit-Learn library, are used. In the experiments, most of the hyper-parameters for machine learning algorithms were set to default. Table I shows the hyper-parameter values for ML algorithms classifiers.

The accuracy definition is crucial since accuracy is an essential criterion for evaluating the efficiency of prediction systems. Accuracy is frequently used to refer to a system's perfect accuracy. However, accuracy can also relate to a class individual accuracy. For researchers working with unbalanced datasets, the definition of accuracy is the average of the accuracies of all classes, which is crucial. In this report, we used K-Nearest Neighbor (KNN) [34], Random Forest (RF) [3], linear support vector machine (SVM-linear) [31], Decision Tree (DT) [30], and Radial basis function (RBF) support vector machine (SVM-RBF) [11] classifiers to classify and detect benchmark CSE-CIC-IDS2018 intrusion detection dataset.

An intrusion detection system should ideally have a 100 percent attack % true-positive rate (TPR) and a 0% false-positive rate (FPR). However, it is difficult to achieve in practice. Table II and Fig. 6 depict the results of these metrics.

The SVM-RBF classification algorithm, as shown in Table II, is the most successful, with a 99.2% accuracy rate. With a 99.1% accuracy rate, the RF classifier algorithm is the second most efficient. Finally, the DT classifier, which had the lowest accuracy rate of 94.2% was applied to the proposed dataset.

With a precision rate of 99.9%, the random forest classifier classification algorithm, as indicated in Table II, is the most successful. The SVM-RBF algorithm is the second most efficient, with a 99.3 % precision rate. Finally, when applied to the proposed dataset, the DT classifier had the lowest precision rate of 79.9%.

TABLE I. MACHINE LEARNING CLASSIFIERS HYPER-PARAMETER VALUES

Algorithm	**Hyper-parameter**
Decision Tree (DT)	criterion='gini', splitter='best', min_samples_split= 2
Random Forest (RF)	n_estimators=1000, criterion='gini', min_samples_split=2, min_samples_leaf=1
K-Nearest Neighbor (KNN)	n_neighbors=3, weights='uniform', leaf_size=30, metric='minkowski'
Support Vector Machine (SVM-linear)	Regularization parameter (C) =1, kernel='linear'
Support Vector Machine (SVM-RBF)	Regularization parameter (C) =1, kernel='rbf'

TABLE II. COMPARISON OF THE RESULTS USING FIVE MACHINE LEARNING CLASSIFIERS

ML algorithm	**Accuracy**	**Precision**	**F1-score**	**TPR**
DT	0.942	0.799	0.865	0.941
RF	0.991	0.999	0.976	1.000
KNN	0.970	0.888	0.926	0.970
SVM-Linear	0.960	0.853	0.905	0.959
SVM-RBF	0.992	0.993	0.979	0.998

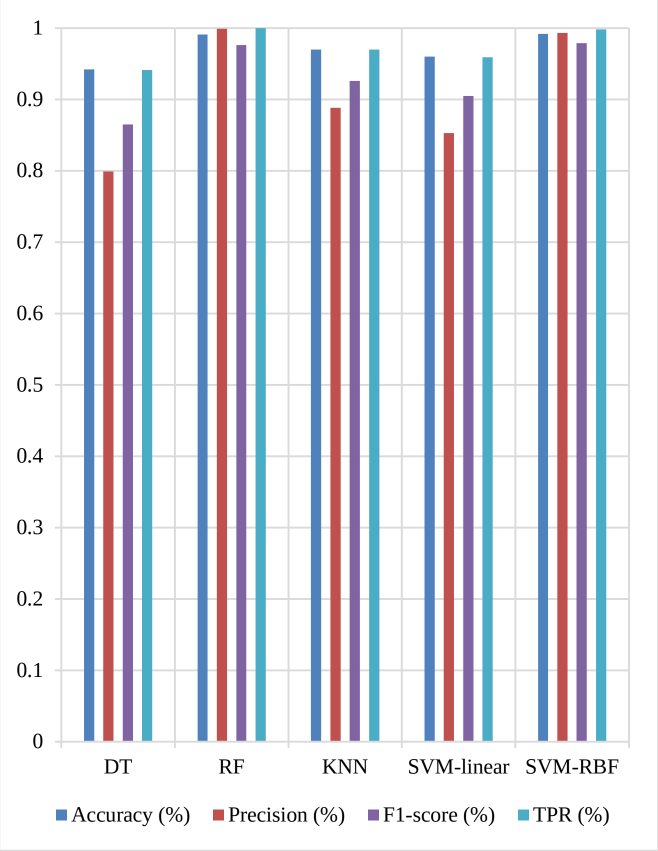

Fig. 6. Performance Analysis of Proposed Framework.

The KNN classifier classification algorithm, as seen in Table II, has the highest recall rate of 96.8%. The second most efficient approach is the SVM-RBF, which has a 96.6% Recall rate. Finally, the DT classifier had the lowest recall rate of 94.3%.

With a 97.9% F1-score rate, the SVM-RBF classification technique, as indicated in Table II, is the most successful. The RF classification algorithm is the second most efficient, with an F1-score rate of 97.6%. Finally, when applied to the provided dataset, the DT classifier had the lowest F1-score rate of 86.5 %.

The RF classification algorithm, as shown in Table II, it is the most successful, with a true-positive rate (TPR) of 100%. With a TPR rate of 99.8%, the SVM-RBF algorithm is the second most efficient. Finally, the DT classifier had the lowest TPR rate of 94.1% when applied to the proposed dataset.

VI. Conclusions and Future Work

The APT attack is not easy or soft kind of attacker. So, detecting it in the early stages will reduce the organization's impact after exploiting it. Also, detecting it using the security exiting tools throw the proposed framework will let it done in a systematic approach. Because of the widespread usage of the Internet in recent years, computational devices can now connect to the universal network from anywhere. However, the anonymous nature of the Internet leads to numerous security flaws in the network, resulting in intrusions. Modern attackers are more intelligent, and they may create new malware and malicious code with the assistance of automated development tools, depending on the limited capability of IDS. This paper uses data transformation and normalization with a reduction procedure using PCA. The benchmark CSE-CIC-IDS2018 dataset is consisted of five different machine learning classifiers for malware IDS detection (DT, RF, KNN, SVM-Linear, and SVM-RBF). The experimental finding showed that the proposed models had a satisfactory performance, specifically when using Random Forest and support vector machine with Radial basis function classifiers, which have a 100% true-positive rate. Several machine learning methods are being transferred to deep learning models due to the convenience of big data technologies. This paper is a preliminary experiment to see how machine learning algorithms can simply and effectively detect attacks from network data traffic. As a result, in the future, deep learning algorithms are recommended to be applied for big DNS data requests.

Acknowledgment

“The author is thankful to the Deanship of Scientific Research, College of Computer and Information Sciences (CCIS) at King Saud University for funding this research.”

References

[1] P. N. Bahrami, A. Dehghantanha, T. Dargahi, R. M. Parizi, K.-K. R. Choo et al., "Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures," Journal of information processing systems, vol. 15, no. 4, pp. 865-889, 2019.

[2] A. S. Ahmed, S. Deb, A.-Z. S. B. Habib, M. N. Mollah and A. S. Ahmad, "Simplistic approach to detect cybercrimes and deter cyber criminals," in 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), 2018, pp. 1-4: IEEE.

[3] S. Li, Q. Zhang, X. Wu, W. Han and Z. Tian, "Attribution classification method of apt malware in iot using machine learning techniques," Security Communication Networks, vol. 2021, 2021.

[4] Q. Zou, X. Sun, P. Liu and A. Singhal, "An approach for detection of advanced persistent threat attacks," Computer Communications, vol. 53, no. 12, pp. 92-96, 2020.

[5] B. Stojanović, K. Hofer-Schmitz and U. Kleb, "Apt datasets and attack modeling for automated detection methods: A review," Computers Security, vol. 92, p. 101734, 2020.

[6] C. Do Xuan, M. H. Dao and H. D. Nguyen, "Apt attack detection based on flow network analysis techniques using deep learning," Journal of Intelligent Fuzzy Systems, vol. 39, no. 3, pp. 4785-4801, 2020.

[7] Y.-x. Xie, L.-x. Ji, L.-s. Li, Z. Guo and T. Baker, "An adaptive defense mechanism to prevent advanced persistent threats," Connection Science, vol. 33, no. 2, pp. 359-379, 2021.

[8] G. Zhao, K. Xu, L. Xu and B. Wu, "Detecting apt malware infections based on malicious dns and traffic analysis," IEEE access, vol. 3, pp. 1132-1142, 2015.

[9] A. L. G. Rios, Z. Li, K. Bekshentayeva and L. Trajković, "Detection of denial of service attacks in communication networks," in 2020 IEEE international symposium on circuits and systems (ISCAS), 2020, pp. 1-5: IEEE.

[10] T. Bodström and T. Hämäläinen, "A novel deep learning stack for apt detection," Applied Sciences, vol. 9, no. 6, p. 1055, 2019.

[11] K. A. A. Alminshid and M. N. Omar, "A framework of apt detection based on packets analysis and host destination," Iraqi Journal of Science, pp. 215-223, 2020.

[12] K. Gopalakrishnan, "Security vulnerabilities and issues of traditional wireless sensors networks in iot," in Principles of internet of things (iot) ecosystem: Insight paradigm: Springer, 2020, pp. 519-549.

[13] S. Mishra, R. Sagban, A. Yakoob and N. Gandhi, "Swarm intelligence in anomaly detection systems: An overview," International Journal of Computers Applications, vol. 43, no. 2, pp. 109-118, 2021.

[14] B. B. Zarpelão, R. S. Miani, C. T. Kawakani and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," Journal of Network Computer Applications, vol. 84, pp. 25-37, 2017.

[15] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSp, vol. 1, pp. 108-116, 2018.

[16] H. Tabrizchi and M. J. T. j. o. s. Kuchaki Rafsanjani, "A survey on security challenges in cloud computing: Issues, threats, and solutions," vol. 76, no. 12, pp. 9493-9532, 2020.

[17] B. Reis, S. B. Kaya, G. Karatas and O. K. Sahingoz, "Intrusion detection systems with gpu-accelerated deep neural networks and effect of the depth," in 2018 6th International Conference on Control Engineering & Information Technology (CEIT), 2018, pp. 1-8: IEEE.

[18] G. Karatas, O. Demir and O. K. Sahingoz, "Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset," IEEE Access, vol. 8, pp. 32150-32162, 2020.

[19] V. Kanimozhi and T. P. Jacob, "Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset cse-cic-ids2018 using cloud computing," International Journal of Engineering Applied Sciences Technology, vol. 4, no. 6, pp. 2455-2143, 2019.

[20] N. F. Haq, A. R. Onik, M. A. K. Hridoy, M. Rafni, F. M. Shah et al., "Application of machine learning approaches in intrusion detection system: A survey," IJARAI-International Journal of Advanced Research in Artificial Intelligence, vol. 4, no. 3, pp. 9-18, 2015.

[21] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," in 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2020, pp. 118-124: IEEE.

[22] Y.-J. Lee, Y.-R. Yeh and Y.-C. F. Wang, "Anomaly detection via online oversampling principal component analysis," IEEE transactions on knowledge data engineering, vol. 25, no. 7, pp. 1460-1470, 2012.

[23] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar and V. Venkatakrishnan, "Holmes: Real-time apt detection through correlation of suspicious information flows," in 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 1137-1152: IEEE.

[24] C.-H. Liu and W.-H. Chen, "The study of using big data analysis to detecting apt attack [j]," Journal of Computer Science, vol. 30, no. 1, pp. 206-222, 2019.

[25] F. J. Abdullayeva, "Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm," Array, vol. 10, p. 100067, 2021.

[26] B. I. Messaoud, K. Guennoun, M. Wahbi and M. Sadik, "Advanced persistent threat: New analysis driven by life cycle phases and their challenges," in 2016 International conference on advanced communication systems and information security (ACOSIS), 2016, pp. 1-6: IEEE.

[27] A. A. Ahmed and Y. W. Kit, "Collecting and analyzing digital proof material to detect cybercrimes," in 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2018, pp. 742-747: IEEE.

[28] D. X. Cho and H. H. Nam, "A method of monitoring and detecting apt attacks based on unknown domains," Procedia Computer Science, vol. 150, pp. 316-323, 2019.

[29] W. Matsuda, M. Fujimoto and T. Mitsunaga, "Detecting apt attacks against active directory using machine leaning," in 2018 IEEE Conference on Application, Information and Network Security (AINS), 2018, pp. 60-65: IEEE.

[30] Y.-H. Kim and W. H. Park, "A study on cyber threat prediction based on intrusion detection event for apt attack detection," Multimedia tools applications, vol. 71, no. 2, pp. 685-698, 2014.

[31] M. Li, W. Huang, Y. Wang, W. Fan and J. Li, "The study of apt attack stage model," in 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1-5: IEEE.

[32] A. Ajibola, I. Ujata, O. Adelaiye and N. A. Rahman, "Mitigating advanced persistent threats: A comparative evaluation review," Int'l J. Info. Sec. Cybercrime, vol. 8, p. 9, 2019.

[33] Y. Zhauniarovich, I. Khalil, T. Yu and M. Dacier, "A survey on malicious domains detection through dns data analysis," ACM Computing Surveys, vol. 51, no. 4, pp. 1-36, 2018.

[34] A. Ajibola, I. Ujata, O. Adelaiye, N. A. Rahman and Cybercrime, "Mitigating advanced persistent threats: A comparative evaluation review," Int'l J. Info. Sec. Cybercrime, vol. 8, p. 9, 2019.