Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.036946
Article

Tech Science Press

Advanced Persistent Threat Detection and Mitigation Using Machine Learning Model

U. Sakthivelu and C. N. S. Vinoth Kumar*

Department of Networking and Communications, College of Engineering and Technology (CET), SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, India
*Corresponding Author: C. N. S. Vinoth Kumar. Email: vinothks1@srmist.edu.in

Received: 17 October 2022; Accepted: 06 December 2022

Abstract: The detection of cyber threats has recently been a crucial research domain as the internet and data drive people's livelihood. Several cyber-attacks lead to the compromise of data security. The proposed system offers complete data protection from Advanced Persistent Threat (APT) attacks with attack detection and defence mechanisms. The modified lateral movement detection algorithm detects the APT attacks, while the defence is achieved by the Dynamic Deception system that makes use of the belief update algorithm. Before termination, every cyber-attack undergoes multiple stages, with the most prominent stage being Lateral Movement (LM). The LM uses a Remote Desktop protocol (RDP) technique to authenticate the unauthorised host leaving footprints on the network and host logs. An anomaly-based approach leveraging the RDP event logs on Windows is used for detecting the evidence of LM. After extracting various feature sets from the logs, the RDP sessions are classified using machine-learning techniques with high recall and precision. It is found that the AdaBoost classifier offers better accuracy, precision, F1 score and recall recording 99.9%, 99.9%, 0.99 and 0.98%. Further, a dynamic deception process is used as a defence mechanism to mitigate APT attacks. A hybrid encryption communication, dynamic (Internet Protocol) IP address generation, timing selection and policy allocation are established based on mathematical models. A belief update algorithm controls the defender's action. The performance of the proposed system is compared with the state-of-the-art models.

Keywords: Advanced persistent threats; lateral movement detection; dynamic deception; remote desktop protocol; Internet Protocol; attack detection

1 Introduction

Advanced Persistent Threats (APT) are one of the major cyber security attacks that have far-reaching consequences on multinational corporations, governments and the public. Attackers must be successfully thwarted from achieving their malicious goals, such as sabotaging a program, infrastructure takeover, credential stealing, etc. This is the ultimate goal of cybersecurity. Xuan et al. [1]

debate the relevance of the advanced or sophisticated nature of the threat while defining APT as a consistent cyber-attack on a target in multiple stages to compromise the organisation by retrieving information, inherently causing a maximal loss in terms of finance and cyber damage. In 2018, the annual loss incurred by cyber-attacks such as APTs were predicted to increase by more than six trillion dollars. Post the North Korean-sponsored attack on Sony in 2014 and a devastating distributed denial-of-service (DDoS) attack on Dyn in 2016, most organisations and enterprises have faced increasing rates of cyber-attacks, especially in the form of APTs [2]. Due to the significant losses incurred, a higher ratio of investments is observed in APT detection and prevention systems.

Usually, the steps of an APT can be characterised by the well-researched reconnaissance of their target to minimise their vulnerability, appropriating a weaponised strategy, widespread usage of malware in lateral movement, followed by data exfiltration from the target organisation using information. Bahrami et al. [3] enlist a cyber kill chain (CKC)-based on seven stages of APT attack that also include reconnaissance, weaponization that mainly includes using phishing, Structured Query Language (SQL) injection, spyware, spam, delivery, exploitation, installation, command, and control (C2), and action on objectives (AoO). The loss of secure information from an organisation, government or commercial can compromise infrastructure and military installations. Corporate and nation/state-sponsored espionage to procure state-of-the-art technology and intellectual property (IP) is also one of the main objectives of APTs. On that note, an APT attack can be handled on two fronts: an apt defence or prevention system and a sufficient attack detection system.

Due to the nature of the APT ransomware as a stealthy threat actor, a good attack detection system should be able to overcome the limitations of the traditional feature-based system by identifying abnormal patterns and attempts in computer networks and correlating them over a long period [4] and differentiating between false positives and false negatives [2]. Similarly, APTs are strategically motivated and well-funded, resulting in a non-repetitive pattern of attacks, which is highly unnoticed by traditional misuse-based detectors that use signature detection patterns of past similar activities [5]. When compared to signature-based detection, anomaly-based detection could be proposed to detect any divergent patterns from normal events in the network (a baseline profile). However, attackers often circumvent the detector by treating network events and system calls as temporal sequences, resulting in underperforming APT detection [6].

Wang et al. [4] classify APT attack detection into two different models based on the host and the network traffic. Host-based detection systems use classification models such as random forest and algorithms such as Naïve Bayes and decision trees to analyse the network connectivity, Central Processing Unit (CPU) usage, memory access, and process creation. Network traffic-based detection collects communication traffic data and analyses it by feature extraction and detection. However, the drawback with the implementation of detection is that it utilises intrusion detection and machine learning. Since attack detection is a fundamental classification issue, neural networks have been observed to deal with attack recognition effectively. For APT attack recognition, a forward feedback neural network model was proposed by Chen et al. [7], integration of support vector machine (SVM) and neural networks, and novel recurrent neural networks (RNN) have also been explored [8,9]. Due to the high complexity and limitations of gradient spread and network layers, real-time APT attacks are often dealt with better by deep neural networks than neural networks in APT attack identification by Ameli et al. [10].

This paper is organised such that the first section details the characteristics of APT along with the type of dataset used and the challenges and opportunities for APT attack detection. A survey of the various APT attacks and measures is detailed in Section 2 while Section 3 outlines the problem formulation for the proposed work. Section 4 details the various attacks detected and the algorithms involved. The defender mechanism is laid out in Section 5 along with its mathematical modelling. Section 6 details the experimental analysis carried out and the results recorded. Based on the observations a conclusion is drawn in Section 7.

Cyberspace is popularly used by several nations, states and governments for carrying out attacks. As a result of these cybernetic skills, disruption in electrical supplies and concerns related to election tampering prevails. Since its discovery, APT attacks have become a vulnerable and damaging aspect where even high-profile systems are easily hacked despite the complex protection algorithms [11]. Categorized under conventional and unconventional, many APT attacks took place in different parts of the world, like China, Pakistan, Ukraine, and so on, related to intellectual property, privacy, and finance [12]. The difference between APT and traditional attacks is given in Fig. 1.

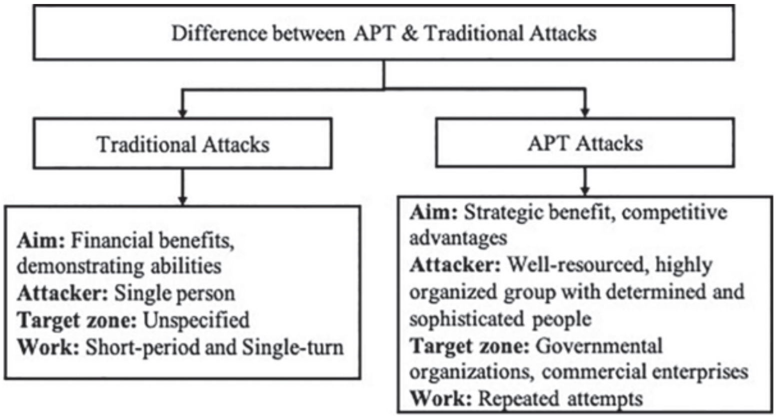

Difference between APT & Traditional Attacks
Traditional Attacks
APT Attacks
Aim: Financial benefits, demonstrating abilities
Attacker: Single person
Target zone: Unspecified
Work: Short-period and Single-turn
Aim: Strategic benefit, competitive advantages
Attacker: Well-resourced, highly organized group with determined and sophisticated people
Target zone: Governmental organizations, commercial enterprises
Work: Repeated attempts

Figure 1: Difference between APT and traditional attacks

To find out whether the occurring attack is an intentional APT or not, certain criteria are mentioned by authors in reference [13–15]. Some of the inferences are given below:

APT attacks can be avoided in several ways: Unexpected or likely assaults require minimum countermeasures and security procedures to prevent such occurrences.

This attack requires slight modification on the portion of attackers: If the attacker's goal doesn't require any modification or evasive movement concerning defensive movements, there occurs an issue in the target's environment.

APT attack uniqueness in its variants: His assault's effectiveness mainly depends on the novel approaches and attack methodologies. However, the established process and tools can detect the attack if the techniques aren't novel.

APT attacks are either specific or broad, as this assault isn't consistent in all the attack categories. They can be segregated into five distinct stages for dealing with the attack. They are as follows:

Stage 1-Reconnaissance: In stage one, the target is clear and becomes more efficient concerning the exploration level. It is a vital stage. *Stage 2-Establishing Foothold:* Stability, entrance and penetration onto the objective occur in this stage. As accessing the target's network is the attacker's primary goal, this stage serves to be the second significant stage. *Stage 3-Staying Undetected:* For stealing the sensitive data and comprising the critical components, the attackers need to traverse the target's network alongside and stay hidden or undetectable. *Stage 4-Impairment/Exfiltration:* Operations such as the delivery of the attacker's command, control centre and data retrieval for obtaining corporate data are done in this stage. This is the stage where the attacker can destroy or weaken the essential components of the target organization. *Stage 5-Post Impediment/Post-Exfiltration:* Accomplishing the attacker's goal, such as deactivating critical components, destroying evidence, exfiltration process completion, and clean withdrawal guarantee from the network's organization, is done in this final stage.

For building an efficient model, the dataset sets a milestone [16]. The commonly used APT datasets are tabulated below in Table 1.

Table 1: Commonly used APT datasets

Reference no.	Name	Year	Description	Opensource/Paid
[17]	UNIBS	2011	In three days of work, the edge router of the University of Brescia was completely traced. For their creation, 20 computers running the GT client domain were responsible.	Paid
[18]	DARPA98	1998	This dataset is a collection of communication between source and destination IP addresses. Darpa consists of a variety of attacks from different IP addresses. It was updated in 2015.	Free
[19]	TRAbID	2017	Collected in a simulated environment enriched with one hundred clients and a Honeypot server. It comprised 16 distinct IDS assessment scenarios, and the test lasted for 30 min.	Paid
[20]	CIC-IDS2018	2018	Includes system and network logs. Over eight days, log data and network traffic are taken. The test was conducted in an environment with 30 servers and 420 machines.	Free

In the APT attack detection process, certain challenges are incurred. In this section, a few are discussed in detail about Challenges and Opportunities for APT Detection.

Long Duration Attacks—Sometimes, APT attacks are performed for a long duration, hence detecting them is quite challenging. If the system shows any suspicious behaviour, it is further correlated with the previous ones in the system. Whereas, the situation is critical if it is a large network with more connections, as false positives and incorrect leads are possible.

Combination with Malware detection—In an APT attack, for establishing communication and data exfiltration tunnels, submission of malware is required where numerous studies were carried out, among which the best example is by Sriram et al. [21], who used an end-to-end deep learning algorithm in identifying different types of malware of dynamic file size.

Powerful & Determined Attackers—The strength and determination of APT attackers is another challenge. Even if a strong defence is in place, it is easy for the attackers to build a complex tool or strategy to break down the defence system. Especially, at present, with the invention and availability of plenty of resources, new malware and custom tools are developed by attackers for attaining their goal.

Lack of Dedicated APT Network Intrusion Dataset—Although the popular datasets mentioned in Table 1 are useful, there still is a need for an efficient network intrusion detection dataset for investigation. For example, KDD 99 is a public benchmark used for evaluating the performance of the system most widely prevalent in studying an IDS network [22]. While both APT and DoS attacks are prevalent in the dataset, for research and analysis, most of the dataset doesn't work out as the host gets compromised gradually. Hence, for reducing the false positives and improving IDS performance, the dataset needs to be labelled properly.

Infrastructure-Oriented Challenges—Another challenge in detecting and preventing an APT attack is the infrastructure or the environment-oriented threats like a large number of correlating events, large interconnections and data exfiltration techniques.

Adversarial ML-Based Attack Detection Methods—Last but not least challenge encountered in the detection of APT attacks is bypassing the defence systems without discovering them. Adversarial Machine Learning (AML) misleads the ML classifiers, and the samples created in the technique, like Fast Gradient Sign and Jacobian-Saliency Map attack, affect the deep-learning-based NIDS. Hence, the effectiveness of AML training needs to be increased.

2 Related Work

In the current Intrusion Detection System (IDSs), the detection of APT is a major challenge, as mentioned before. Numerous research studies were carried out to address this MSA attack and assault. A novel host-based APT detector–"SPuNgae", was proposed by authors in reference [23] that monitors the network and finds out the malicious URLs. Similarly, for the detection of data exfiltration, Sigholm et al., in reference [24], utilised Data Leakage Prevention (DLP) algorithm, which looks on for data leakage in the network. By employing Cyber Counterintelligence (CCI) sensors, the location of leaked data is detected accurately. Another approach was presented in [25] called TerminAPT, which tracks the data flow taking place in the APT campaign. In an APT system, for gathering information regarding Point of Entry (PoE), Spear phishing is the common method. In [26], the authors discussed the methodology involving mathematical computational analysis techniques in picking out spam emails. Developed to identify tokens and characters of spam behaviour, tokens need to be defined thoroughly in the algorithm, which is the only limitation. Different APT detection & prevention models and ML-based APT methods are also available and discussed in upcoming sub-sections. The various methodologies carried out so far have several drawbacks that need to be addressed and this work aims to improve some of these aspects of it. The proposed work addresses the drawbacks of the previously existing work in terms of accuracy, precision, F1 score and recall recording.

Various detection and prevention approaches were developed by renowned cyber security researchers for handling APT attacks that are state-of-art and tactical. In reference [27], the Honey-pot technique employable at the production part of the network is given. By diverting adversaries' focus from the intended goal, the technique possesses drawbacks like lack of real-time APT detection & prevention and post-infiltration detection. Big Data Analytics [28], a recent trend creator in the world of technology and data science, is recognised to be a good technique for detecting APTs. For pattern matching, this method is also quite helpful in analysing the flow of network topology. Unlike the Honeypot method, Big Data analytics have disadvantages like non-protection in real-time and false positives. Finally, a Context-based framework [29] improves user experience in information systems related to medical or context-based systems.

Data labelling is crucial in generating the correct answer when the question arises without certainty. In this process, the Machine Learning technique gets the primary focus. In ML, three types are available: Supervised, Unsupervised, and Semi-supervised. In detecting malicious RDP, supervised and unsupervised ML algorithms are preferred.

Supervised ML Algorithms–Supervised ML algorithms effectively detect anomalies. Some of the popular techniques under this category are Logistic Regression (LR), Decision Tree (DT) classifier, Classification and Regression Tree (CART) [30], LogitBoost (LB) [31], and LightGBM [32].

Unsupervised ML Algorithms—In this category, the algorithm tends to learn the infrastructure of input data without the necessity of explicit labels. Clustering algorithms come under this type where K-means [33], Agglomerative clustering, Density-based Spatial Clustering for Applications with Noise (DBSCAN) [34], and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [35] get the high focus.

3 Attack Detection and Problem Formulation

There is a considerable limitation in the internet event log datasets that depict the behaviour of a real user. The datasets are taken from Windows event log datasets from Los Alamos National Laboratory (LANL). Network-based intrusion detection is facilitated by most of the publicly available datasets. Sensitive information in the host event logs limits organisations from distributing such data. This limitation is overcome by simulating the behaviour of users and attackers and generating synthetic datasets. The user behaviour in a real-world environment may not be completely depicted in such approaches as the datasets are generated purely based on hypothetical assumptions. Datasets are significant for successfully training and testing any machine learning algorithm. The primary limitations of generic intrusion datasets and systems are as follows:

The attack traffic is captured at the external endpoints. When the attack vectors of the APTs are within the internal networks, these datasets are ineffective.

The APT attacks from sophisticated attackers may not be represented as the distinction between normal and anomalous behaviour is surmised in these datasets.

In semi-supervised learning, the real-world settings are not reflected efficiently, leading to data imbalance. However, supervised models operate optimally.

Real-time detection and prevention systems are limited, and most existing systems work on post-infiltration scenarios.

3.1 Dataset Combination

A comprehensive and unified dataset is combined with preserving user behaviour's realistic nature.

Comprehensive dataset—The Operationally Transparent Computing Cyber (OpTC) dataset from Defense Advanced Research Projects Agency (DARPA) is a comprehensive dataset used for anomaly detection. The host-based telemetry records are available in this dataset, and it is the most detailed public dataset that currently exists. Such datasets are privately gathered by executing professional security services or cybersecurity operations within the organization. Despite being generated by simulators, this dataset offers different types of malicious engagements similar to the baseline activity of modern tactics. The complexity and structure of private datasets are replicated for cyber defence research. The Los Alamos National Laboratory's (LANL) Unified Host and Network dataset is another public dataset that has similar features to the OpTC dataset.

Unified Dataset—For quantitative comparison, reproduction and further research, several barriers are significantly lowered by the data format in unified datasets. However, in the Transparent Computing program, the Engagement 3 and 5 datasets are the only publicly available unified datasets for detecting APT attacks. Self-collected limited attack data is used in most of the existing research work. These datasets do not entirely represent the real-world sophisticated attack scenarios as they contain limited attack scenarios. The LANL-based unified dataset is collected for 90 days. Detailed event logs on windows are provided comprehensively in this dataset, along with the missing logoff events. Events are categorized into days despite the obfuscation of timestamps in the dataset. However, the activities of benign users are only available in this dataset.

3.2 The Lateral Movement Detection Algorithm

Certain limitations are observed when the comprehensive and unified datasets are used individually. To overcome such limitations, malicious data is injected from the comprehensive dataset into the unified dataset. The attack event patterns and properties are retained as both datasets are gathered within the same organization. However, the mismatch and variations in the hash functions make it challenging to merge two datasets. To avoid bias in classification using machine learning, the existing hosts can be mapped into a larger group of hosts in the new dataset. In the comprehensive dataset, the collection of malicious logon events is termed M, and in the unified dataset, the collection of benign RDP logon events is termed B.

The source host S_{mi} is mapped for each event $e_i \in$ M to S_{mj}, which is a unique source host that is randomly selected from an event $e_j \in$ B. For the event e_i, $\{U_i, D_i\}$ represent the host tuple user name and destination mapped to $\{U_k, D_k\}$, a randomly selected unique tuple from $e_k \in$ B. The modified malicious events' e_i' is inserted into the benign dataset chronologically and labelled. Algorithm 1 provides the details of the injection of malicious remote desktop protocol authentication events. μ represents the mean of the benign session duration, σ^2 represents the variance of the benign session duration, x is the set of benign source hosts, and y is the set of malicious source hosts.

3.3 Defense Mechanism

In a real-time environment, the network defender takes action as soon as the attacker progresses through the system and limits this progress. For this purpose, a dynamic deception model is introduced that uses socket synchronization and IP address generation. These steps use hybrid encrypted communication and block cipher symmetric encryption, respectively. A Hidden Markov Model (HMM) is used for timing selection. The dynamic host configuration protocol (DHCP) enables policy allocation. The action of the defender is controlled by a belief update algorithm. A joint probability distribution

Algorithm 1: Malicious Remote Desktop Protocol authentication events injection

```
Initialize: $\mu$, $\sigma^2$, x and y/malicious and benign variables
Malicious_AuthTuple ← (“username” + “destination event”)/in benign
Benign_AuthTuple ← (“username” + “destination event”)/in malicious
/Dictionary mapping malicious and benign data from source
Source ← dict{}
    for each host ∈ y
        Source[host] ← x.randomPop()
End
/Dictionary mapping malicious and benign data from the tuple
AuthTuple ← dict{}
    for each tuple ∈ Malicious_AuthTuple do
        AuthTuple[tuple] ← Benign_AuthTuple.randomPop()
End
/Rewrite malicious event fields and insert them in benign
for each event ∈ Malicious, do
new source ← Source[event.Host]
newuser ← AuthTuple[event.Tuple]username
newdestination ← AuthTuple[event.Tuple]destination
session ← GaussianRandom($\mu$, $\sigma^2$)
modified ← newEvent
append.Benign(modified)
End
return Benign
```

is used over the attacker types, and security states and the capability of the attacker are captured using a belief matrix. The defender can decide whether to save or spend more resources to thwart the reconnaissance missions based on the type of attacker.

3.4 Socket Synchronization and IP Address Generation

Based on User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), socket communication can be further divided into two communication methodologies. It is easier to synchronise sockets and hence is chosen as the optimal means in this work. Since the socket doesn’t have a fixed port, attackers find it difficult to attack. Fig. 2 represents the communication flow of the encryption. A hybrid end-to-end encryption communication module is designed based on the original socket communication technology. The message M is extracted as plaintext using ‘H’ as hash algorithm. This is followed by signing the hash value SM2 and the package II is further processed using the SM4 hashing algorithm. At the receiving end, the sender’s public key is used to recover the plaintext and hashing is again carried out with SM4.

A total of 32 rounds of non-linear iteration is carried out with the help of key expansion and packet encryption with SM4 grouping. Thus, a pseudo-random sequence is generated and can be further incorporated as the dynamic IP address table. An encryption model using cipher block chaining is used wherein the previous round of encryption operation is XORed with the initial parameter. Here, the key generated is generated and used as the seed key input.

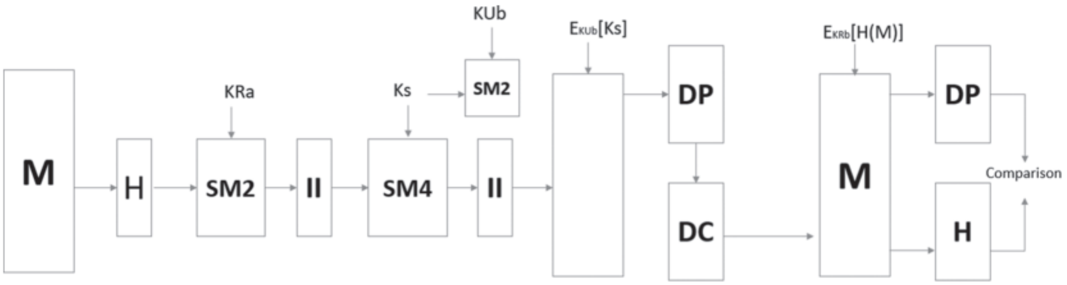


Figure 2: Encryption and identity process

4 Experimental Analysis

4.1 Hidden Markov Model

A Hidden Markov model (HMM) is incorporated to identify the future state of the system. Here the Markov chain is capable of determining the transition probability based on the visible states such that q_t represents the current state and q_{t+1} is the future state. For N states, discrete observation symbols, $\overline{O}_m = (\overline{o}_1, \overline{o}_2, \overline{o}_3, \overline{o}_4, ..\overline{o}_{M-1}, \overline{o}_M\}$ and $S = \{s_1, s_2, s_3, s_4, ..s_{N-1}, s_N\}$, the Hidden Markov model can be represented with initial matrix π_i, observation emission matrix $c = \{c_i(\overline{O_k})\}$ such that $i, j \epsilon [1, 2, 3 \ldots, N]$ and $k \epsilon [1, 2, \ldots.M]$. HMM, can be described using Eq. (1).

$$\lambda = (A, B, \pi) \tag{1}$$

Fig. 3 represents the dynamic policy assignment of the proposed work. As shown in the figure, the dynamic policy has two major parts: dynamic IP address table and dynamic time. As the first step, a dynamic policy is generated by the dynamic deception domain, which forwards it to the DHCPv6 server depending on the honeypot, Intrusion detection system and firewall's information. The server uses an IPv6 address dynamic protocol to control the lease period, which impacts the IP address. A dynamic timing generation is triggered by the length of the lease period.

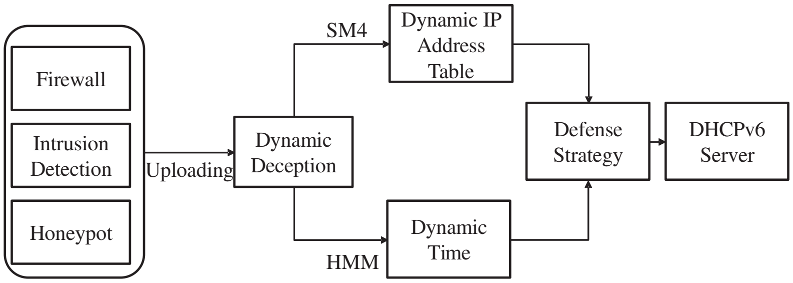


Figure 3: Dynamic policy allocation

4.2 Update Algorithm

A heuristic search algorithm called the online defence algorithm is used for identifying defence actions in real-time. An online defence algorithm built on the sample generates security alerts on detecting an attacker's progress via the network using the security model structure, paving the way to large-scale domain computing analysis. Blocking vulnerability or a similar defence action is employed to analyze the progress in assessing the attacking path of the attacker. The challenge lies in optimal computing action while deceptively interacting with the attacker, as far as scalable networks are concerned. The offline POMDP solver evaluates the optimal action for every belief state before runtime. Even though the solver has higher efficiency, its ability to capture the optimal action will be impossible in the case of large networks. Zainudin et al. in [16] addressed this issue with the help of Partially Observable Monte-Carlo Planning (POMCP), which can handle a large-scale network.

Compared with offline methods, online methods skip execution and computation stages, resulting in a more scalable approach. Action nodes and belief nodes are the two types of nodes in POMCP.

Action nodes are the children's nodes of the belief state that can be reached using actions.

The belief state denotes belief nodes.

In this proposed methodology, a POMCP algorithm similar to the selection process is used along with the solution to large observation space problems with a modified belief update procedure. In this technique, Algorithm 2 is used to analyze if every incoming alert $z_i \in Z$ matches with Z(s) = Z(e), which is the security state. When the attacker triggers an attempt to exploit, the alert is generated. On the other hand, alerts that are not in A(s) will not be generated. Hence these alerts are declared as false alerts. A generative model is called at the initial stage of simulation to provide cost, observation and sample success for a particular state and action $(s, \varphi, y, -) \sim G(s, \varphi, u_r)$. Here the state-action pair is represented as $\propto_t$.

Algorithm 2: Belief Update Algorithm

```
Initialize: $n_k$, $\propto_{t+1} = U_{a(r,f)}$, added_num=0
procedure Belief_Update ($\propto_t, u_r, y_r$)
        while added_num $< n_k$ do
    $(s, \varphi) \sim \propto_t$
    $(s, \varphi, y, -) \sim G(s, \varphi, u_r)$
if $y^{Z(s)} = y_r^{Z(s)}$ then
        $\propto_{t+1} \leftarrow \propto_{t+1}\ U\,\{s', \varphi'\}$
        added_num$\leftarrow$added_num+1
```

Using successive sampling and generative models, the history of the search tree can be built, as shown in Fig. 4. In a search tree, history is represented by the nodes, while the branches that extend from the nodes denote possible future histories. On the other hand, the greedy tree policy is followed by the MCTS such that the highest value is chosen at the initial stage of simulation.

4.3 Mathematical Modeling

Consider the attack arrival from the user b to host a at time t,

$$A_{ab}[t] = \sigma_b \cdot 1_{\{b\in N_a[t], b\notin N_a[t-1]\}} \tag{2}$$

Here, σ represents the attack load, and N represents the set of hosts infected under the specific attack. The rate at which the host receives a response is given by:

$$r_a[t] = \sum\nolimits_{b\in N_a[t]} \sigma_b \cdot r_{ab}[t] \tag{3}$$

Characterization of the host dropping rate is done by the following expression while considering the attack loads:

$$d_a[t] = \sum\nolimits_{b\in N_a[t]} \sigma_b \cdot d_{ab}[t] \tag{4}$$

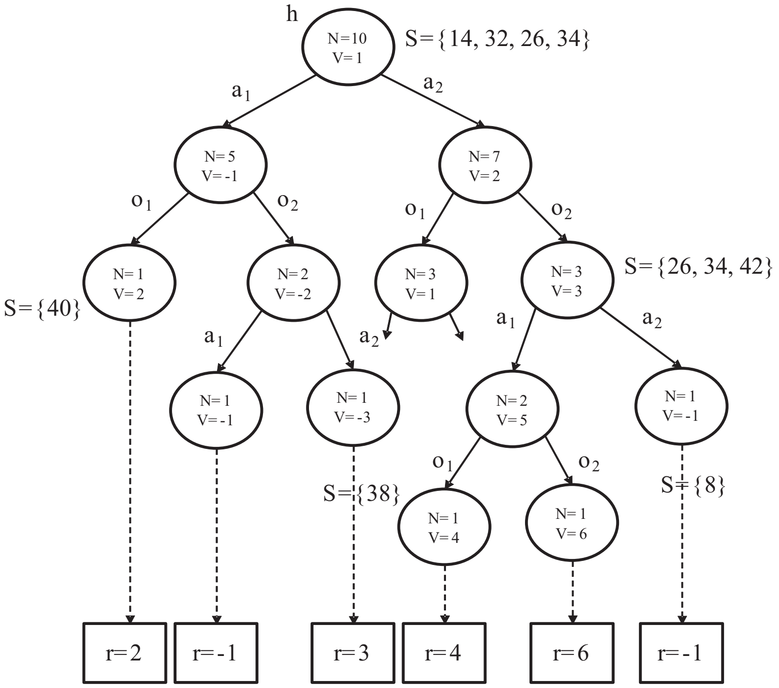


Figure 4: POMDP environment search tree construction with real action and real observation o

For each attack event b towards the host a, the dynamic risk level is modelled using the following expression:

$$L_{ab}\,[t] = \omega_b e^{t-t_b} \cdot 1_{\{b \in Na[t]\}} \tag{5}$$

Here, ω_b represents the risk score. The time at which host b is compromised is estimated by $t - t_b$. Preference ranking and maximum response guarantee of the host are estimated to obtain high response efficiency under stable security conditions. The preference ranking of host a is given by

$$R_a\,[t] = \{R_a^1, \ldots\ldots, R_a^{|Na[t]|}\} \tag{6}$$

for the risk level vector

$$l_a\,[t] = \{l_{ab}\,[t]\,|a \in N_a\,[t]\}. \tag{7}$$

Further, the end-to-end packet delay is modelled by the following expression considering the processing overhead, queueing delay, propagation delay, transmission delay and processing delay that occurs while ensuring confidentiality, integrity and authentication.

$$D_{total} = N[D_{prov} + D_{qu} + D_{tr} + D_{prpg} + D_{proc}) \tag{8}$$

When interacting with the attacker, the defender's optimum action is POMDP. This can be defined using the equation:

$$\begin{aligned} V^{\pi}\,(b_0) &= \sum\nolimits_{t=0}^{\infty} \gamma^t c(b_t, u_t, \varphi_t) \\ &= \sum\nolimits_{t=0}^{\infty} \gamma^t E[c(b_t, u_t, \varphi_t)|b_0, \pi] \end{aligned} \tag{9}$$

such that $0 < \gamma < 1$ lies between 0 and 1, while $c\left(b_t, u_t, \varphi_t\right)$ denotes the cost of the state.

The delay-aware virtual queue Q_a is updated using the following expression:

$$Q_a\left[t+1\right] = \max\{Q_a\left[t\right] - r_a\left[t\right] - d_a\left[t\right] + \epsilon_a 1_{\{Q_a[t]>0\}}, 0\} \tag{10}$$

4.4 Evaluation Metrics

A cluster of nodes is considered for performing pre-processing, visualization and data analysis. Intel Xeon E-2224G Processor 4.70 GHz CPU and 32 GB RAM are used for this purpose. 10 Gbps Ethernet is used for interconnecting the nodes. Microsoft Azure VM is used for training and validating the model under supervised learning conditions. Unsupervised learning is performed on Intel Xeon W-2200 Processor with 18 AVX-512-enabled cores and 1 TB DDR4 memory. Data visualization is performed using Grafana, and the dataset is ingested into the Datadog cluster by deploying an instance. Pandas, SciPy, NumPy and such Python packages are used for data pre-processing. Keras and MLlib libraries are used for developing the ML models in Python.

4.5 ML Metrics

Various ML techniques are evaluated, and their performance metrics are estimated to define the malicious RDP sessions.

$$Accuracy = \frac{TP + TN}{Total\ subjects} \times 100\% \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{12}$$

$$F1\ score = 2 \times \frac{TP}{TP + FN} \tag{13}$$

$$Sensitivity/Recall = \frac{TP}{TP + FN} \times 100\% \tag{14}$$

$$AP\ score = \sum\nolimits_n \left(Recall_n - Recall_{n-1}\right) \times Precision \tag{15}$$

$$Sepcificity = \frac{TN}{FP + TN} \times 100\% \tag{16}$$

$$GMeean = \sqrt{Sensitivity + Specificity} \tag{17}$$

TP is True Positive, TN is True Negative, FP is False Positive, and FN is a False Negative value.

4.6 Experiment

K-cross fold validation is employed to validate the machine learning models with baseline features, as in Table 2. The value of k is considered to be 10. Comparison is made among the Random Forest (RF), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Feed-forward Neural Network (FNN), Decision Tree Classifier (DTC) and Adaptive Boosting (AdaBoost) classifiers. The AdaBoost classifier offers better accuracy, precision, F1 score and recall. This is because these classifiers are designed to boost the performance of the existing classifiers. Fig. 5 represents the performance of the AdaBoost classifier in terms of the metrics for a different number of clusters. Fig. 6 shows the cross-validation results for various iterations. It compares the proposed cross-validation model along with existing models such as robustness tests [36] and bootstrapping [37].

Various attack types are used with multiple levels of stealth, aggression and attack knowledge ranging between high, moderate and low [38,39]. The attack and success probabilities for each condition are estimated. Further, the performance of the proposed model on individual datasets and the combined dataset is compared in Fig. 7 for various parameters. Better classification performance is observed when the combined dataset is tested with malicious traces from a user.

Table 2: Estimation of performance metrics during detection of RDP session with ML classifiers

Classifier	Accuracy	Precision	F1 score	Recall
RF	99.8%	99.6%	0.96	96.0%
LR	98.4%	10.7%	0.03	1.3%
GNB	99.4%	86.3%	0.85	83.1%
FNN	96.6%	0%	0	0%
DTC	99.9%	99%	0.95	92.6%
AdaBoost	99.9%	99.9%	0.99	99.8%

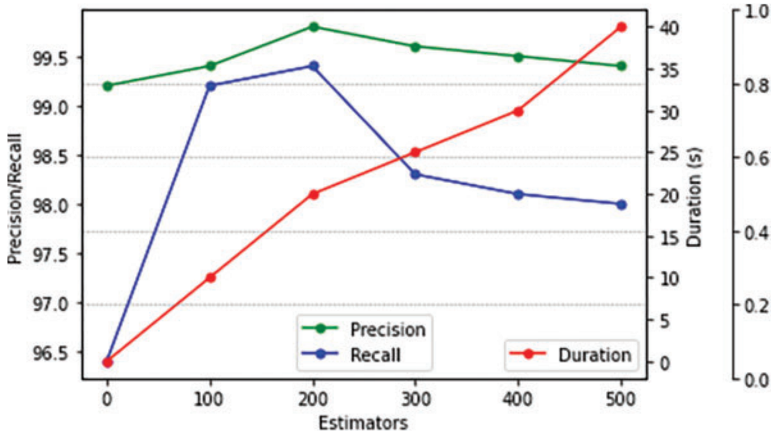


Figure 5: Number of estimators *vs*. precision, recall and training duration for stand-alone AdaBoost model

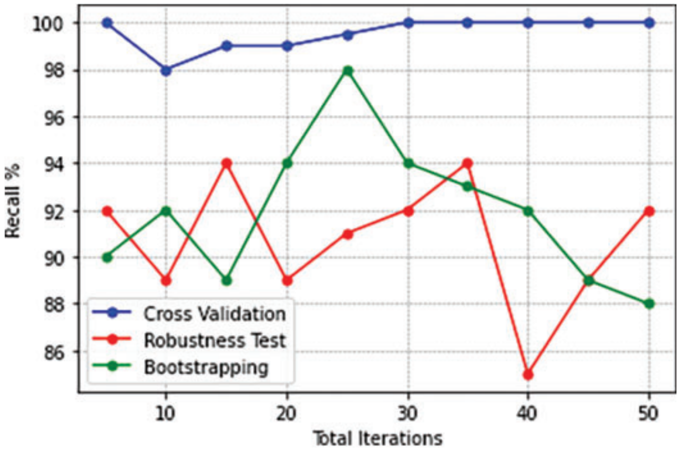
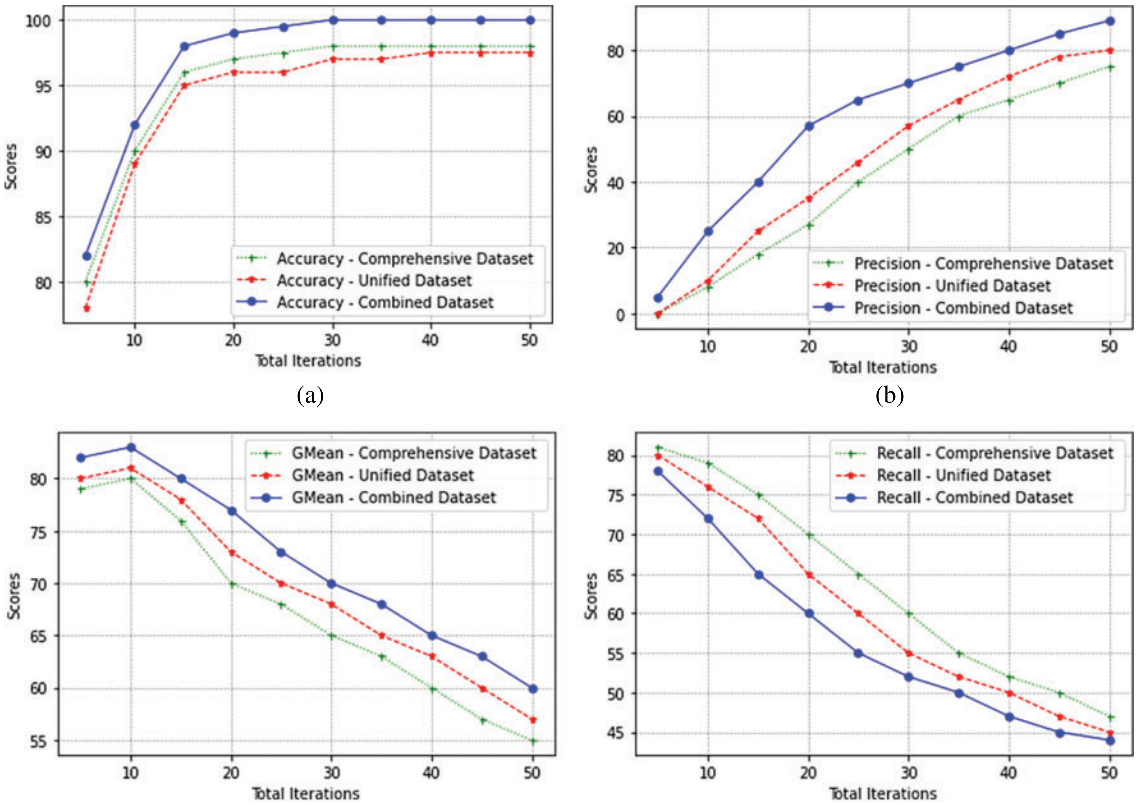


Figure 6: Comparison of recall value during cross-validation for various iterations



(a) (b)

(c) (d)

Figure 7: Performance evaluation on testing with independent and combined datasets

Ensemble ML can be used for consolidating stand-alone classifiers to further improve performance. Here, a majority voting algorithm is used for leveraging the ML models in the ensemble. A conservative approach termed weighted voting is used, where weights are assigned based on intuition. The false positives and negatives may be reduced by assigning a higher weight to the classifier with better performance. A series of experiments are conducted to analyse the impact of adversarial attacks on the proposed model. Based on these experiments, it is evident that the proposed model is robust and successfully detects and mitigates various types of adversarial attacks.

5 Conclusion

In recent years, cyber threats serve as a severe threat to people using the internet extensively for all purposes. Advanced persistent threats (APTs) are one of the most complex attacks which last a long time. During the APT attack and its lateral movement stage, a common tool that can be used to prevent the attack from intervening is the RDP. In this work, malicious attacks in RDP are detected and mitigated by leveraging the event logs in Windows. Multiple datasets are combined to overcome the shortcomings of the individual datasets while remaining faithful to the attack models. The anomalous RDP sessions are detected using a supervised learning algorithm by extracting relevant features. Classification algorithms namely Random Forest (RF), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Feed-forward Neural Network (FNN), Decision Tree Classifier (DTC) and Adaptive

Boosting (AdaBoost) are evaluated and compared for precision, recall, F1 score and accuracy. It is found that the AdaBoost classifier offers better accuracy, precision, F1 score and recall recording 99.9%, 99.9%, 0.99 and 0.98%. The dynamic deception model is used as a defence mechanism. It is a combination of the block cipher symmetric encryption, HMM, DHCP and a belief update algorithm. Future work is directed towards the deployment of the system in online learning, hybrid systems and other session-based protocols to test the performance. Further, more event logs and test scenarios can be implemented to improve efficiency.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. D. Xuan, D. Duong and H. X. Dau, "A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic," *Journal of Intelligent & Fuzzy Systems,* vol. 40*,* no. 6*,* pp. 11311–11329, 2021.

[2] I. Ghafir, H. Mohammad, P. Vaclav, H. Liangxiu, H. Robert *et al.,* "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems,* vol. 89*,* no. 4*,* pp. 349–359, 2018.

[3] 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.

[4] X. Wang, Q. Liu, Z. Pan and G. Pang, "APT attack detection algorithm based on spatio-temporal association analysis in industrial network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2020, no. 12, pp. 1–10, 2020.

[5] Z. Li, X. Cheng, L. Sun, J. Zhang and B. Chen, "A hierarchical approach for advanced persistent threat detection with attention-based graph neural networks," *Security and Communication Networks*, vol. 2021, pp. 1–14, 2021.

[6] A. Khalid, A. Zainal, M. A. Maarof and F. A. Ghaleb, "Advanced persistent threat detection: A survey," in *The Proc. of IEEE 3rd Int. Cyber Resilience Conf. (CRC)*, Langkawi Island, Malaysia, pp. 1–6, 2021.

[7] Y. Chen, S. Kar and J. M. F. Moura, "Dynamic attack detection in cyber-physical systems with side initial state information," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4618–4624, 2016.

[8] K. Wang, M. Du, Y. Sun, A. Vinel and Y. Zhang, "Attack detection and distributed forensics in machine-to-machine networks," *IEEE Network*, vol. 30, no. 6, pp. 49–55, 2016.

[9] A. Czajka and K. W. Bowyer, "Presentation attack detection for iris recognition: An assessment of the state-of-the-art," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

[10] A. Ameli, A. Hooshyar, E. F. El-Saadany and A. M. Youssef, "Attack detection and identification for automatic generation control systems," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 4760–4774, 2018.

[11] N. Falliere, L. O. Murchu and E. Chien, "W32. Stuxnet dossier," *Symantec Security Response*, vol. 1, no. 4, pp. 1–69, 2011.

[12] K. Xing, A. Li, R. Jiang and Y. Jia, "A review of APT attack detection methods and defense strategies," in *The Proc. of IEEE 5th Int. Conf. of Data Science and Cyberspace*, DSC, Hong Kong, China, pp. 67–70, 2020.

[13] A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges and research opportunities," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[14] I. Jeun, Y. Lee and D. Won, "A practical study on advanced persistent threats," *Computer Applications for Security, Control and System Engineering*, vol. 339, pp. 144–152, 2012.

[15] V. Prenosil and I. Ghafir, "Advanced persistent threat attack detection: An overview," *International Journal of Advanced Computers and Networks*, vol. 4, no. 4, pp. 50–54, 2014.

[16] Z. S. B. Zainudin, "A case study of advanced persistent threats on financial institutions in Malaysia," Msc Thesis, International Islamic University Malaysia, 2017.

[17] UNIBS, "UNIBS," 2011. [Online]. Available: http://netweb.ing.unibs.it/

[18] R. P. Lippmann, R. K. Cunningham, D. J. Fried, I. Graf, K. R. Kendall *et al.,* "Results of the DARPA 1998 offline intrusion detection evaluation," MIT Lincoln Laboratory, 1999.

[19] TRAbID, "TRAbID," 2017. [Online]. Available: https://secplab.ppgia.pucpr.br/trabid

[20] CIC-IDS2018, "CIC-IDS2018," 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[21] S. Sriram, R. Vinayakumar, V. Sowmya, M. Alazab and K. P. Soman, "Multi-scale learning based malware variant detection using spatial pyramid pooling network," in *Proc. of IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, pp. 740–745, 2020.

[22] M. S. Al-Daweri, K. A. Z. Ariffin, S. Abdullah and M. S. E. Md. Senan, "An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system," *Symmetry*, vol. 12, no. 10, pp. 1666, 2020.

[23] M. Balduzzi, V. Ciangaglini and R. McArdle, "Targeted attacks detection with SPuNge," in *Proc. of IEEE 11th Annual Conf. on Privacy, Security and Trust*, Tarragona, Spain, pp. 185–194, 2013.

[24] J. Sigholm and M. Bang, "Towards offensive cyber counterintelligence: Adopting a target-centric view on advanced persistent threats," in *The Proc. of European Intelligence Security Informatics Conf. (EISIC)*, Uppsala, Sweden, pp. 166–171, 2013.

[25] G. Brogi and V. V. T. Tong, "Terminaptor: Highlighting advanced persistent threats through information _ow tracking," in *The Proc. of 8th IFIP Int. Conf. on New Technologies, Mobility and Security (NTMS)*, Larnaca, Cyprus, pp. 1–5, 2016.

[26] J. V. Chandra, N. Challa and S. K. Pasupuleti, "A practical approach to E-mail spam_lters to protect data from advanced persistent threat," in *Proc. of Int. Conf. on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, India, pp. 1–5, 2016.

[27] H. Bari, "Protecting an enterprise network through the deployment of honeypot," Bangladesh University, Post Graduate Thesis, 2021.

[28] A. A. Cardenas, P. K. Manadhata and S. P. Rajan, "Big data analytics for security," *IEEE Security & Privacy*, vol. 11, no. 6, pp. 74–76, 2013.

[29] P. Giura and W. Wang, "A context-based detection framework for advanced persistent threats," in *Proc. of ASE Int. Conf. on Cyber Security*, Alexandria, VA, USA, pp. 69–74, 2012.

[30] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, "Classification and regression trees. The Wadsworth statistics/probability series," in *Wadsworth & Brooks, Cole Advanced Books & Software*, Monterey, CA, 1984.

[31] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[32] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen *et al.,* "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1–9, 2017.

[33] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc.of the Fifth Berkeley Symp. on Mathematical Statistics and Probability*, Oakland, CA, USA, vol. 1, pp. 281, 1967.

[34] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A density based algorithm for discovering clusters in large spatial databases with noise," *KDD-96 Proceedings*, vol. 96, no. 34, pp. 226–231, 1996.

[35] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.

[36] T. Bai, H. Bian, A. A. Daya, M. A. Salahuddin, N. Limam *et al.,* "A machine learning approach for RDP-based lateral movement detection," in *Proc. of IEEE 44th Conf. on Local Computer Networks (LCN)*, Osnabrueck, Germany, pp. 242–245, 2019.

[37] D. Tychalas, A. Keliris and M. Maniatakos, "LED Alert: Supply chain threats for stealthy data exfiltration in industrial control systems," in *Proc. of IEEE 25th Int. Symp. on On-Line Testing and Robust System Design (IOLTS)*, Rhodes, Greece, pp. 194–199, 2019.

[38] A. A. Movassagh, J. A. Alzubi, M. Gheisari, M. Rahimi, S. K. Mohan *et al.,* "Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model," *Journal of Ambient Intelligence Humanized Computing*, 2021. https://doi.org/10.1007/s12652-020-02623-6

[39] O. A. Alzubi, J. A. Alzubi, M. Alazab, A. Alrabea, A. Awajan *et al.,* "Optimized machine learning-based intrusion detection system for fog and edge computing environment," *Electronics*, vol. 11, no. 19, pp. 3007, 2022.