Future Generation Computer Systems 106 (2020) 501–517

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics

Aaron Zimba [a], Hongsong Chen [a,*], Zhaoshun Wang [a], Mumbi Chishimba [b]

[a] Department of Computer Science and Technology, University of Science and Technology Beijing, 100083, China
[b] Department of Information Technology, National Institute of Public Administration, Lusaka, 10101, Zambia

ARTICLE INFO

Article history:
Received 20 June 2019
Received in revised form 25 December 2019
Accepted 20 January 2020
Available online 22 January 2020

Keywords:
Targeted attacks
Attack network
Command and control
Vertex degree
Clustering coefficient

ABSTRACT

Advanced Persistent Threats (APT) present the most sophisticated types of attacks to modern networks which have proved to be very challenging to address. Using sophisticated attack techniques, attackers remotely control infected machines and exfiltrate sensitive information from organizations and governments. Security products deployed by enterprise networks based on traditional defenses often fail at detecting APT infections because of the dynamic nature of the APT attack process. To overcome the current limitations of attack network dynamics faced in APT studies, an innovative APT attack detection model based on a semi-supervised learning approach and complex networks characteristics is proposed in this paper. The entire targeted network is modeled as a small-world network and the evolving APT-Attack Network (APT-AN) as a scale-free network. Finite state machines are employed to model the state transitions of the nodes in the time domain in order to characterize the state changes during the APT attack process. The effectiveness of the model is demonstrated by applying it to real-world data from a large-scale enterprise network consisting of 17,684 hosts from the Los Alamos security lab. The proposed approach analyzes efficiently the large-scale dataset to reveal APT attack characteristics between the command and control center and the victim hosts. The final result is a ranked list of suspicious hosts participating in APT attack activities. The average detection precision of three APT stage is 90.5% in our proposed APT detection framework. The results show that the model can effectively detect the suspicious hosts at different stages of the APT attack process.

1. Introduction

Advanced Persistent Threats (APTs) are a class of targeted attacks [1] that are launched by highly skilled technical threat actors who are well funded by big corporations or nations [2]. Unlike conventional attacks, APTs are human-driven and are carried out over a long period of time after surveillance of the targeted network [3–5]. They leverage a wide spectrum of attack vectors for infiltration not limited to social engineering and exploitation of both known and zero-day vulnerabilities [6,7].

The economic impact of APT attacks has been documented across different sectors [8,9] and has not only been limited to financial losses but to reputation and legal lawsuits as well [10]. Such incurred losses span millions of dollars [11]. As such, APTs are a security menace that most information systems cannot manage to ignore.

Traditional defense systems have not been very effective in defending against APTs because APTs exhibit characteristics that are difficult to detect [12]. APT attacks mimic the normal behavior of a system both in terms of generated system calls and the network traffic thereof [13,14]. Furthermore, APTs evade defense systems using polymorphic software code and bypass perimeter firewalls using standard protocols and permitted ports [15]. Although the use of polymorphic or encrypted code in malware has seen security solutions eschew signature-based detection techniques for heuristic-based detection, it is still difficult to detect APTs because of their unpredictable non-repetitive behavior. And unlike automated traditional malware, APTs compromise only a few hosts in a target network which might comprise hundreds or thousands of hosts [16]. Such a huge number of hosts generate enormous traffic which inadvertently makes detection difficult because the noise generated by APT activities is very small.

All these characteristics of APTs make them stealthy [5] and thus require a dynamic detection approach in order to capture the aforesaid. Considering this, detection and analysis of APTs have drawn significant attention from both the industry and

[*] Corresponding author.
E-mail addresses: azimba@xs.ustb.edu.cn (A. Zimba), chenhs@ustb.edu.cn (H. Chen), zhswang@sohu.com (Z. Wang), mumbi.chishimba@gmail.com (M. Chishimba).

https://doi.org/10.1016/j.future.2020.01.032
0167-739X/© 2020 Elsevier B.V. All rights reserved.

academia. However, the dynamic growth of the APT attack network (APT-AN) which evolves as the attack progresses, is an attribute seldomly considered in APT attacks studies [4,5,17–20].

The main goal of this paper is to identify a few susceptible hosts from a network of tens of thousands of hosts suspected to be participating in APT activities based on network statistics. The scope does not extend to explicitly finding and pinpointing a host compromised by an APT. Instead, this paper endeavors to detect and classify a group of hosts exhibiting suspicious APT-related activities. As such, this provides an avenue for security personnel to concentrate their efforts on a few hosts thereby improving efficiency. A holistic approach to detection of APTs is infeasible [4,21]. However, in large scale APT-related activities, groups of hosts exhibit general complex network behaviors [22]. It is on this premise that the proposed modeling and detection methodologies are based.

To overcome the current limitations of attack network dynamics faced in APT studies, a complex network analysis model is proposed to model the evolution of the underlying APT-AN and use finite state machines to model the state transitions of the nodes with respect to the time domain. To further address the dynamicity, modeling of the entire targeted network using the small-world networks model (see [23]) and the underlying evolving APT-AN using the scale-free networks model (see [24]) is proposed. Network statistics are not limited to Domain Name System (DNS) [25] communications with the command and control (C2) centers and network flows are extracted. Attention is focused on communications with the C2s because apart from data exfiltration, recent APTs attacks have sought to sabotage targeted networks on the cloud [26] via denial-of-resource attacks [27] (DoR) and ransomware attacks costing millions of dollars in incurred losses [28]. In these attacks, communication with the C2 is of paramount importance.

As such, a framework is proposed that is capable of capturing and analyzing enormous volumes of network traffic data and detect APT-related activities with regards to C2 communications and data exfiltration.

The remainder of the paper is organized as follows: Section 2 presents the background and related works. The motivations for this paper are presented in Section 3 where the challenges in detecting APTs are discussed. Section 4 presents the methodology and data while the proposed framework is presented in Section 5. Results and discussion are brought forth in Section 6 and the conclusion is drawn in Section 7.

2. Background and related works

With the rise of cyber-attacks, literature [22,29–31] has in the recent past seen several works seeking to formalize the APT problem related to intrusion detection and prevention. In this section, 3 main related categories in current literature are considered; APT attack modeling, APT attack detection, and APT insider threat detection.

2.1. APT attack modeling

In [32], the authors propose an APT model based on classical Petri nets. The attack object is extracted from a state-space set where the transition node-set in partitioned into attack intention set, environment condition, penetration expansion, and vulnerability utilization. The resultant is a 6-tuple model that expresses state transitions for a combination of state space, attack process, and scene. However, the model is not flexible as it is limited to 8 states and neither does it consider the growth of the underlying APT-AN in the targeted network.

In [33], the authors present a comprehensive study of APT attack modeling in cloud computing. The authors address the APT challenge of characterizing interlinked attack paths generated by APT attackers upon the exploitation of vulnerabilities exhibited in cloud components by chaining marginal and conditional probabilities to characterize the multiple attack paths from the attack source to the target node. They evaluate the likelihood of an APT occurring in a given attack path and propose an optimized algorithm to find the shortest attack path from multiple sources. They employ Bayesian networks to formulate the shortest APT attack paths but leave detection to future works.

In [34], the authors propose a multi-phase transferable Markov model where the APT exfiltration phases are mapped to a cyber kill chain. Probability is used to measure the impact exerted on the target network by the next attack action. The behavior of the attacker is expressed as a 3-tuple comprised of branches set, nodes set and the branches' prior belief. As such, the model provides an avenue for the network administrator to detect the early stages of an APT. However, the model does not capture the APT-AN which can change through different states not considered therein.

In [35], authors present an APT detection methodology that uses low-level interception and correlates the events of the operating system with those of the network based on the semantic relationships that they define between the entities in system ontology. In their scheme, the suspicious malicious events, especially those that implicitly violate the security policies, are generated and detected based on the event relations and defined security policies. As such, they use state changes in security policies to detect suspicious APT activities. Their approach is limited to Windows-based operating systems and is not scalable to large networks.

In [22], the authors propose a machine learning-based system which seeks to predict and detect APT attacks in a systematic way. They develop eight methods to detect different techniques used during the various APT steps. From here, they develop a correlation framework to link the outputs of the proposed detection methods, which aims to identify alerts that could be related and belong to a single APT scenario. Finally, a machine learning-based prediction module is proposed based on the correlation framework output. However, the approach lacks internal detection which corresponds to the lateral movement stage of the proposed model. Nonetheless, the authors add that if their system were to monitor the internal network traffic, other detection modules could be added to detect brute force and other attacks, thereby increasing the detectable steps of the system. Furthermore, some modules of their system were only validated on simulated data.

In [36], the authors establish a theoretical framework to deduce the characteristics of an information-based APT attack on a given internal network. They devise a mathematical framework that includes the initial entry model for selecting the entry points into the attack network and a targeted attack model for studying the intelligence-gathering phase, strategy decision-making phase, weaponization, and lateral movement phase. Using a series of simulations, the authors find the optimal candidate nodes in their initial entry model, after which they observe the dynamic change of the targeted attack model and thus verify the characteristics of the APT attack under consideration. However, the model is not applied to real-world data and it also does not consider the modeling of data exfiltration in the last phase of an APT attack process, which is one of the most important aspects of APT attacks [37].

The FireEye's Mandiant [38] APT attack lifecycle is an 8-phase process that describes the various stages of sophisticated cyber-attacks comprising of Initial Reconnaissance — the stage where the attacker conducts research on a target, surveils the network ad gathers information for use in future attacks, Initial Compromise — the stage where the attacker successfully executes malware on one or more hosts in the targeted systems, Establish Foothold — the stage where the attacker establishes and maintains continued control over recently compromised systems, Escalate Privileges — the stage where the attacker obtains greater access to systems and data the thereof through various means such as password hash dumping, Internal Reconnaissance — the stage where the attacker explores the compromised system in order to gain a better understanding of the environment, Move Laterally — the stage where the attacker utilizes his access to traverse from system to system within the compromised environment, Maintain Presence — the stage where the attacker ensures persistent access to the environment via methods such as backdoor implantation, Complete Mission — the stage where the attacker accomplishes his ultimate goal which is usually exfiltration of data from the target environment.

However, some stages in this model are unrealistic to detect, such as Initial Reconnaissance where the actions of this stage are undeterminable and beyond the security analyst's domain.

The APT intrusion kill-chain by Hutchins et al. [39] is a 7-phase systematic process to target and engage an APT adversary to create desired effects. The process is comprised of Reconnaissance — research stage where identification and selection of targets is done, Weaponization — integrating remote access malware and exploitation of vulnerabilities in the target system, Delivery — transmission of the weaponized malware to the target environment via a specified infection vector, Exploitation — the stage where the weaponized malware is triggered on the targeted victim, Installation — installing a remote access malware such as Rootkit or Trojan for establishment of a persistent presence, Command and Control — establishment of a line of connection with the servers of the APT human actors, beaconing and acquisition of further directives, Actions on Objectives — the end-goal which is usually exfiltration of targeted data to the C2 servers or any other destination servers.

2.2. APT attack detection

In [40], the authors propose a practical framework for the detection of APT data exfiltrations via analyzing APT network communications. Like many detection techniques, this approach based on automatic signatures has two major shortfalls: Assumption that the APT adversary uses clear-text communication which is to be matched to their generated signatures. The major hurdle in this approach is that APTs tend to use encrypted communications [41]. The second assumption that all data of interest is known a-priori so as to generate required signatures for matching data exfiltration. This is not a realistic undertaking as it is practically infeasible to tell beforehand what data the APT adversary will be after. In [42–44], the authors propose the detection of data exfiltrations using host-based data logs. Inasmuch as such might be practical to small networks, it is ineffective to networks as those considered in this paper comprising tens of thousands of hosts as it would require huge amounts of data from each host. Moreover, the works do not give an insight into the underlying APT-AN. Chen et al. [45] present a very comprehensive study on APT attacks with the characterization of the attack model and the commonly used analysis techniques with specific case studies. A very clear distinction is given between traditional threats and APT. Based on the "intrusion kill chain" [46], the authors further detail how APTs work and formulate a typical attack process comprising 6 phases namely (i) reconnaissance and weaponization (ii) payload delivery (iii) initial intrusion (iv) command and control (v) lateral traversal (vi) data exfiltration. Like many other works, there is no consideration on the APT-AN which is pivotal in the formulation of strategies for thwarting further network surveillance upon APT compromise. Furthermore, there is no application to real-world intrusion data. In [21], authors use real-world network data to formulate an APT detection mechanism but likewise do not consider the APT-AN and the dynamicity thereof. Though the authors in [47] consider the dynamism of APT-AN, their application to real-world intrusion data is not extensive and application to large-scale networks is limited. They run an experiment where the target node which is an application server resides in a class A private IP address space. The attacker resides in a Class C private IP address subnet who compromises hosts in the class A subnet and elevates privileges upon further exploitation and surveilling the hosts seeking to authenticate with the compromised server.

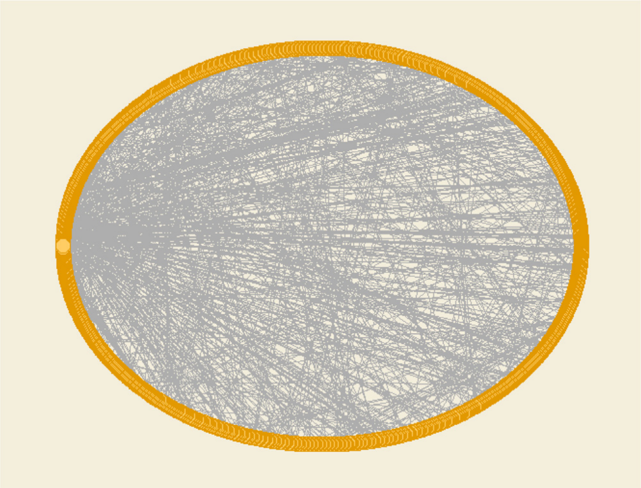


Fig. 1. High-density network structure with abundant hubs of the Los Alamos DNS dataset.

2.3. Small-world model in APT communication network

Small-World (SW) networks are characterized by large clustering coefficients $C(p_i)$ and a significantly small mean shortest-path length $L(p_i)$. The large C_i results into over-abundance of hubs in the communication network as depicted from the Los Alamos data-set in Fig. 1 generated in Pajek [48]. As such, the clustering coefficient of the network according to the Watts–Strogatz (WS) model [23] is defined as:

$$C\left(p\right)=\frac{3\left(K-1\right)}{2\left(2K-1\right)}\cdot\left(1-p\right)^{3} \tag{1}$$

where k is the vertex degree and p the connectivity probability. The vertex degree k denotes the number of existent edges of a node whereas p is the probability of linking the node with another. The clustering C_i coefficient denotes the cliquishness of the network which depicts deviations from the randomness of the entire network structure.

The yellow dots denote nodes of the network and the gray links denote the edges between nodes. As can be seen, the network has many hubs and a visibly high clustering coefficient. This is because users in an enterprise network tend to access almost the same resources. Furthermore, even though cloud providers might offer load-balancing through different servers, user access patterns will be iterative and not greatly differ because of the use of load-balancing algorithms such as round-robin on the server-end. The average path length as per the WS model is given as:

$$L\left(p\right)=\frac{2N}{K}\cdot f\left(\frac{2N}{K}\cdot p\right) \tag{2}$$

where N is the size of the network and with the function $f(x)$:

$$f(x) = \begin{cases} C, x \ll 1 \\ \dfrac{\ln x}{x}, & x \gg 1 \end{cases} \tag{3}$$

It is visible from Eqs. (2) and (3) that the small and short path lengths in Fig. 1 are mediated by the abundant hubs in the network. The abundant hubs have, as shown in Fig. 1, a larger fraction of vertices with high degrees. Consequently, the node degree distribution is fat-tailed. The vertex degree k entails the number of nodes a vertex is connected to both internally and externally whilst the clustering coefficient C_i entails the connectivity, hence the trust relationships of the neighboring nodes. As such, the APT can exploit a fraction of nodes from k and their relationships depending on C_i to traverse the network and exfiltrate data where applicable.

Although $L(p)$ denotes the average path length between nodes, APTs have a tendency of not taking obvious routes in a network to avoid detection. Therefore, the relevance of $L(p)$ will be dependent on the given attack scenario and the intention/motivation of the attacker.

2.4. Scale-free APT-AN network model

The presence of many clusters and a higher vertex degree enables the APT to traverse the network after infiltration thereby generating an APT-AN. In the beginning, the would-be nodes of the APT-AN are isolated. In this case, $C_i = 0$ and $k_i = 0$. As such, there is no linkability in the APT-AN but the nodes are only susceptible to exploitation depending on the exhibited vulnerability. The attacker discovers vulnerable nodes n_{i+1} via reconnaissance or surveillance probes in the network which are directly reachable from the current node n_i. This implies that the APT-AN grows dynamically with respect to time. All these activities correspond to the infiltration phase of the proposed APT lifecycle as depicted in Fig. 1 for which $k \neq 0$. As the APT-AN dynamically grows with an increase of C_i and k_i, the capabilities of the APT likewise increase thus transitioning the APT-AN to the lateral movement, C2 beaconing and exfiltration phase. As such, at any given time t_i of the attack process, the nodes of the APT-AN can be partitioned into three active states namely $\boldsymbol{S}_1$, $\boldsymbol{S}_2$ and $\boldsymbol{S}_3$ which correspond to the three stages of the proposed APT lifecycle. The dynamic growth of the APT-AN with respect to time t_n is thus depicted as a time-dependent multi-slice network with transitional states $\boldsymbol{S}_n$. This is shown in Fig. 2.

One major characteristic of this network is the addition of new nodes with respect to time t_n as the attack progresses, a typical characteristic of a dynamic complex network.

Unlike the small-world networks, the addition of these nodes is dependent on the exploitability of the vulnerability exhibited in the node n_i. The vulnerability in a node is determined by many different parameters, thus the probability of adding a new node to the network is not uniform. As such, the APT-AN characterizes a scale-free complex network. The probability of adding a new node to an existent node n_i with vertex degree k_i as per the Barabási–Albert (BA) model [49] is given by:

$$\Pi_i = \frac{k_i}{\sum_{j=1}^{N} k_j} \tag{4}$$

The topology of the APT-AN at t_0 is a sparse unconnected network. The susceptible nodes are not yet traversable due to the attacker's limited knowledge about the communication network. This is representative of the state $\boldsymbol{S}_0$. After a reconnaissance probe, the attacker establishes which nodes are reachable and infects them with APT malware, and this is representative of the state $\boldsymbol{S}_1$, an ideal virtual (dotted) network reflecting the traversability of an omniscient attacker. Traversing the susceptible nodes, which in essence is exploiting the vulnerability in the nodes, transitions the APT-AN to state $\boldsymbol{S}_2$. The topology of the resultant network is determined by the clustering coefficient and the vertex degree. The average clustering coefficient $\langle C \rangle$ as per the BA model is given by:

$$\langle C \rangle = \frac{n^2(n+1)^2}{4(n-1)} \cdot \left[\ln\left(\frac{n+1}{n}\right) - \frac{1}{(n+1)} \right] \cdot \frac{(\ln t)^2}{t} \tag{5}$$

In the same manner, the average path length of the resultant network is given by:

$$L \sim \frac{\ln N}{\ln(\ln N)} \tag{6}$$

where N is the size of the network. Following Eqs. (5) and (6), the average path length, though larger than that of an SW network, is not a very large value and neither is the clustering coefficient. When $\langle C \rangle = 0$, the resultant topology is a star network. If $\langle C \rangle = 1$, the resultant topology is a fully connected mesh network. However, these extreme scenarios are highly improbable in real-world networks because the star topology implies that all nodes in the network exhibit one and only one vulnerability and are only exploitable from the current node. The full mesh topology implies that all nodes in the network are exploitable and traversable from any other node in the network. Since these two scenarios are unrealistic, a scale-free APT-AN for a clustering coefficient in the range $0 < C < 1$ is thus adopted. An adjacency matrix $A_G(\boldsymbol{S}_n)$ of the 9th order is used to depict traversability of the illustrative APT-AN network in Fig. 2 as it grows dynamically through the states $\boldsymbol{S}_1$ to $\boldsymbol{S}_3$ (see Fig. 2a):

It is clear from the matrices that a transition from state $\boldsymbol{S}_1$ to $\boldsymbol{S}_2$ sees the APT-AN grow by five nodes in the x–y plane with the initially infected node n_1 having the highest vertex degree $k(n_1) = 4$ whilst all other connected nodes have $k = 1$. The elements of the matrices are identical about an all-zero major diagonal entailing the absence of self-loops in the APT-AN. This is plausible considering that APTs avoid re-traversing the same nodes to avoid detection. A transition from state $\boldsymbol{S}_2$ to $\boldsymbol{S}_3$ sees the dynamic addition of more nodes to the APT-AN where the vertex degree of node n_3 is $k(n_3) = 4$. Furthermore, state the $\boldsymbol{S}_3$ includes red nodes in the $x - -y - t$ dimensional space. These nodes denote the malicious domains which are the destinations of the data exfiltrated in the last stage of the APT attack lifecycle.

Even though $k(n_1) = k(n_3) = 4$ in state $\boldsymbol{S}_3$, only n_1 characterizes the isthmus of the network considering that it is the infection point of the network. As such, during mitigation upon discovery of compromised nodes, elimination of the isthmus (node n_1 in this case) should be prioritized as it would break the linkability of the APT-AN. Securing a vulnerable node translates to eliminating it from the APT-AN and such, its removal eliminates the edges connected to such a node. This precipitates a reduction in the average vertex degree k.

Since the dynamic growth (addition and elimination) of the APT-AN from state $\boldsymbol{S}_n$ to $\boldsymbol{S}_{n+1}$ occurs with time t_n, Eq. (9) which depicts the attack graph can be modified to reflect the state of the network at $t+1$ as:

$$G_{APT-AN}(t+1) = (N_{t+1}, E_{t+1}) \tag{7}$$

Since a node can be added to the APT-AN upon the exploitation of a vulnerability or be eliminated upon mitigation of a vulnerability, the dynamic state of the APT-AN as it transitions from state $\boldsymbol{S}_n$ to $\boldsymbol{S}_{n+1}$ after time $t+1$ can be expressed as:

$$\boldsymbol{S}_n(t) \longrightarrow \boldsymbol{S}_{n+1}(t+1) : G_{APT-AN}(t+1) = \begin{cases} (N_t \bigcup n^{+}_{t+1}) - (n^{-}_{t+1}) \\ \cdot \\ (E_t \bigcup e^{+}_{t+1}) - (e^{-}_{t+1}) \end{cases}$$

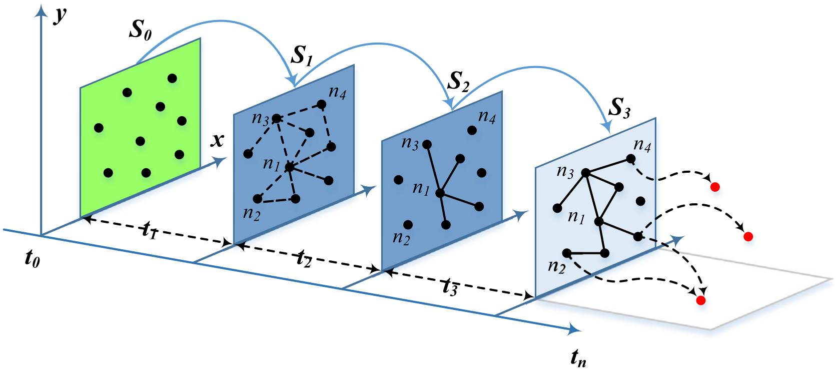


Fig. 2. The dynamic growth of an APT-AN.

$$\begin{tikzcd}
\mathbf{A_G}(\mathrm{S}_1) \arrow[r, bend left] & \mathbf{A_G}(\mathrm{S}_2) \arrow[r, bend left] & \mathbf{A_G}(\mathrm{S}_3)
\end{tikzcd}$$

$$\begin{bmatrix}
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0
\end{bmatrix}
\quad
\begin{bmatrix}
0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0
\end{bmatrix}
\quad
\begin{bmatrix}
0&0&1&0&0&1&1&1&0\\
0&0&0&0&0&0&1&0&0\\
0&0&0&1&1&0&0&1&0\\
0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Fig. 2a.

(8)

whereafter time $t+1$: n_{t+1}^{+} is node addition upon the exploitation of a vulnerability and n_{t+1}^{-} is node elimination upon security mitigation. The added edge e_{t+1}^{+} denotes an increase in the vertex degree associated with n_{t+1}^{+} whereas e_{t+1}^{-} denotes the disappearance of an edge upon elimination of the node n_{t+1}^{-} which reflects a decrease in the vertex degree. The evolution of the APT-AN discussed thus far describes the entire network change and not the state changes of the individual hosts. This leads to the modeling of state changes of the nodes as the APT-AN transitions through the states $\boldsymbol{S}_n$.

This proposed modeling approach comes from the pre-experiment and data analysis. The targeted networks are modeled as a small-world so that the APT attacker can control the targeted network easily. The APT-ANs are modeled as a scale-free so that the APT-AN can increase quickly.

3. APT detection challenges and motivations

3.1. Unpredictability of APT lifecycle stages

Unlike conventional attacks, APTs are multi-stage attacks [50] that spend over long periods while adopting a slow-and-low approach to the severity of the attack resulting in many stages of the APT attack lifecycle. A holistic approach to detection likewise is practically infeasible [5,22,35].

To address the challenge of holistically addressing incompatible and interleaving stages, a compact APT lifecycle based on the APT kill chains presented in [38] and [39] is proposed. The resultant APT attack lifecycle is shown in Fig. 3. The 8 phases of FireEye's Mandiant kill-chain are similar to the 7 phases of Hutchins et al. [46]. The first 3 stages from both models are compacted into the Infiltration phase where the attacker acquires enough details from the target and delivers the malware which is ready to transition the attack process to the next state. The Lateral Traversal stage corresponds to the next 4 stages of FireEye's Mandiant kill chain and the next 2 stages of phases of Hutchins et al. The C2 Beaconing and Exfiltration phase corresponds to the last stage of FireEye's kill-chain and last 2 stages of Hutchins et al. As such, a compact APT lifecycle is proposed which is used to model APTs attack using complex networks and finite state machines in the proceeding section.

3.2. Dynamism of APT-ANs and communication networks

Considering the multi-steps of the APT attack process in Fig. 3, it is clear that after infiltration the attacker consecutively compromises a series of hosts in the target network. Since he can traverse the target network through these hosts, they inadvertently form a sub-network of compromised nodes that is henceforth term Advanced Persistent Threat Attack Network (APT-AN). Since the actions of the attacker are unpredictable, addition of new nodes to the APT-AN likewise is unpredictable. This produces a dynamic non-trivial topological structure that can be modeled as a complex network.

From the analyses that were carried out on real network data from about 17K hosts, it was observed that the resultant network is a dynamic complex network. The growth of the networks is unpredictable since network access for each user is unpredictable. The nodes of the network shown in Fig. 4(a), (b) and (c) from the Los Alamos dataset [51] represent hosts within the target network and their destinations whilst the edges represent communications between the nodes. The DNS dataset is about 1 GB

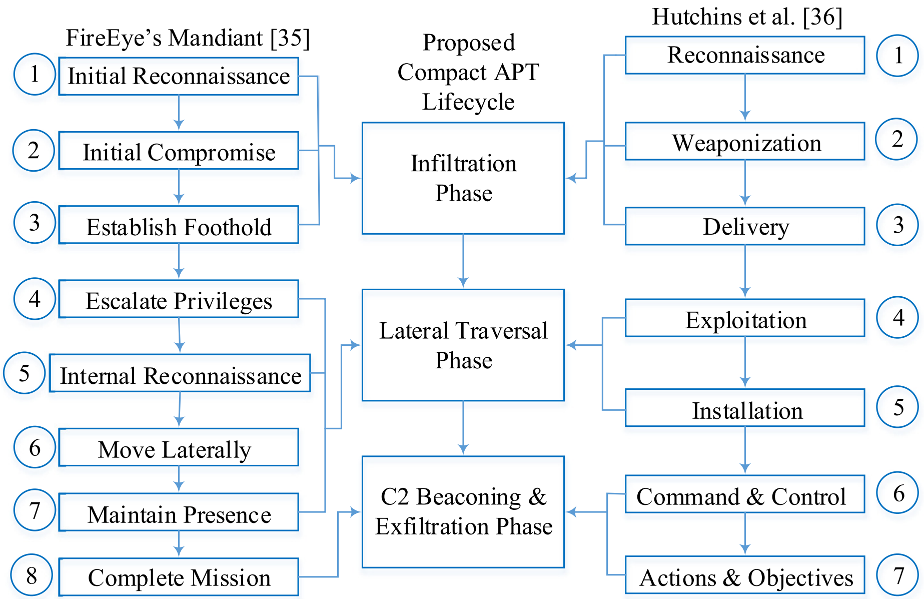
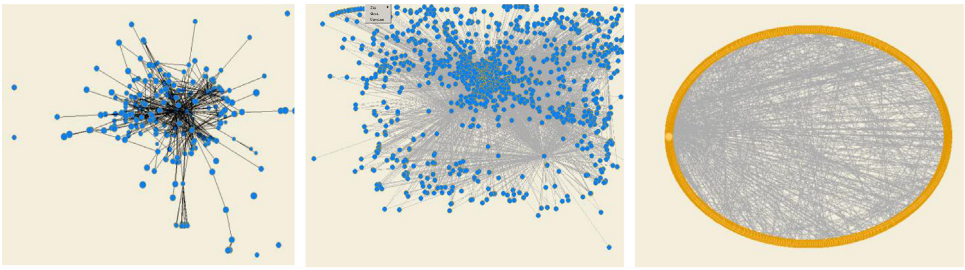
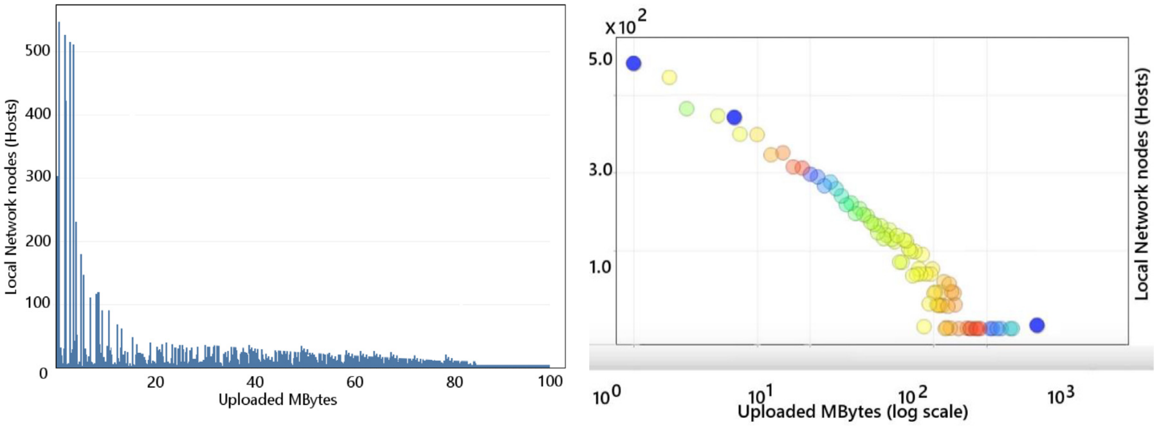


Fig. 3. The proposed compact lifecycle of an APT.



(a) After 1 hour (b) After 12 hours (c) After 24 hours

Fig. 4. The dynamic growth of the communication network with time of the Los Alamos DNS dataset.



(a) Normal linear scale (b) Logarithmic scale

Fig. 5. Distribution of flow records between internal and external hosts of the Los Alamos dataset.

of external and internal name lookup events collected from the Los Alamos DNS servers from within the internal local network.

To address the challenges posed by the network dynamics of the communication network and the APT-AN, the former is modeled as a small-world network and the underlying APT-AN as a scale-free network. Attention is focused on communications with the C2s because apart from data exfiltration, recent APTs attacks have also sought to sabotage targeted networks on the cloud [26] via denial-of-resource attacks [52] (DoR) and ransomware attacks [28].

3.3. Imbalanced data distribution

Large enterprise networks generate huge volumes of network traffic which consist of a very small fraction of APT activities.

Table 1
Overview of the LANL dataset.

Data element	Event count	User count	Computer count
Authentication	1,051,430,459	12,418	17,666
Processes	426,045,096	10,097	11,960
DNS	40,821,591	N/A	15,296
Network flows	129,977,412	N/A	12,027
Red Team	749	98	305
Total	1,648,275,307	12,425	17,684

Table 2
The dataset of the DNS logs.

SN	Time (s)	Source host	Destination host
1	31	C161	C2109
2	35	C5642	C528
3	38	C3380	C22841
N	…	…	…

APT activities are hidden in weak signals of this enormous data source. Furthermore, APTs contact C2s usually in the last stage of the attack cycle but before that, they laterally traverse the network amongst local hosts for varying periods of time. This presents another challenge in that if detection is focused on C2 communication, the APT-AN will continue to grow undetected until such a time when it will be optimal to exfiltrate data. Still more, such local traversal represents a very small fraction of the data generated by local communications. Additionally, most network statistics follow the Power-Law [53] distribution where a large portion of the data is skewed. This is evident in the dataset as shown in Fig. 5 for both the linear and logarithmic scale where the uploaded data implies contact with an external host.

The graphs are heavily-tailed distributions of the Power-Law form. Coarse-grained lustering-based detection techniques would fare poorly in such a dataset. Thus, a combination of a number of features from the dataset for fine-grained detection techniques is adopted.

3.4. Scarcity of public APT data

The classical problem of the lack of publicly available data in the security industry is echoed in the fight against APTs. Due to legal implications and potential loss of reputation among other reasons, most victims of APT attacks do not make their security logs publicly available. To address this challenge, real network data by Los Alamos National Lab (LANL) [51] is used which was captured over 2 months comprising 17,684 hosts to demonstrate the feasibility of the proposed methodology. The dataset also includes APT activities simulated by domain experts (Red Team).

4. Methodology and data

The Los Alamos dataset is approximately 12 GB of compressed data across the 5 data elements; Authentication, Processes, DNS, Network Flows and Red Team. The statistical breakdowns of these elements are shown in Table 1.

Since this work is primarily focused on detection of APT activities in the three stages of the APT attack chain ($\boldsymbol{S}_1$, $\boldsymbol{S}_2$ and $\boldsymbol{S}_3$), data elements relevant to this task are thus selected. To this effect, the use of network traffic which constitutes DNS logs and network flows is adopted. The features of the data elements in both the DNS and network flows datasets were anonymized to preserve the security of the network from which the dataset was derived.

All the data in the DNS and network flows datasets start with a time resolution of one second (1 s). Table 2 shows the properties of the anonymized DNS logs dataset from which features for APT detection are drawn. When an infected a host is initially infected with APT malware in stage $\boldsymbol{S}_1$, it tries to communicate with the C2 server via DNS requests to download instructions and any other related tasks. As such, variations in DNS requests will enable us to detect name resolutions to suspicious domains and destinations.

Network flows will enable us to capture any communications between any infected hosts and a potential victim within the network during stage $\boldsymbol{S}_2$ of the APT attack lifecycle. This is achieved by using a host detected in stage $\boldsymbol{S}_1$ as a seed and applying belief propagation to see which internal hosts it will try to communicate with. To detect data exfiltrations, detected hosts from stage $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are used in this stage. Table 3 shows the characteristics of the anonymized network flow dataset from which features for APT attack detection are drawn.

From Tables 1, 2, and 3, network features are chosen, which are characteristics of dynamic complex networks. These features will enable us to compute the vertices' degree and clustering coefficients to be used in the detection process. This is shown in Table 4.

The Red Team data represents the number of compromise instances during the two months period. Since hosts in the Red Team data are technically labeled as malicious, this data is used in the classification portion of the proposed algorithm. Table 5 shows an expansion of the Red Team dataset from Table 1.

The four parameters in Table 5 namely Time, User ID, Source Host ID, and Destination Host ID are used to map and identify the corresponding host in Tables 2 and 3. These two Tables are used to extract the associated vertex degrees and clustering coefficients of the hosts to determine which stage of the APT attack process the host belongs to. Subsequently, the host in question is labeled as S_1, S_2 or S_3. Since the Red Team dataset consists of compromise incidents, none of the data elements is labeled $\boldsymbol{S_0}$ as this state represents a secure state. As such, the output labels of the mapping are S_1, S_2 or S_3 based on the APT characteristic features of the associated attack phase.

5. Framework of the proposed model

This section provides an overview of the proposed approach. This also includes the tools, methods, and datasets used. Furthermore, the modeling approach for both the APT-AN and the communication network is henceforth formulated. Attack graphs are adopted to be used describing the properties of the networks and their nodes and edges. Nodes represent the different hosts in the network while edges represent the interactions between the nodes. Such interactions are not limited to DNS communications, data transfers and specific net-flows in general. The edges can be weighted or unweighted depending on the selected type of network interactions. Therefore, the graph representative of the general network is defined as:

$$G_{net} = (N_i, E_i) \tag{9}$$

where G_{net} is the generated graph, N_i are nodes such that $n \in N_i$, E_i are edges such that $e \in E_i$ and $i = 1, 2, 3, \ldots, n$. Whether the network under consideration is an APT-AN or the communication network, communication between two nodes N_i and N_j is in two modes; local communication and external communication. In the former, a host communicates locally for services such as file services such as SMB, printing, local domain logins, and so forth. In the latter, a host communicates to an external resolved name such as SaaS service, remote access services, websites, and so on.

APTs hide in both the local and external network communications. In local communications, the APTs usually carry out the infiltration and lateral movement phases activities (*cf.* Fig. 1). In external communications, the APTs carry out C2 beaconing

Table 3
The network flows events dataset.

SN	Time (s)	Duration (s)	Src. host	Src. port	Dst. host	Dst. port	Protocol	Packet count	Byte count
1	1	9	C3090	N10471	C3420	N46	6	3	144
2	1	9	C3538	N2600	C3371	N46	6	3	144
3	2	0	C4316	N10199	C5030	443	6	2	92
N	…	…	…	…	…	…	…	…	…

Table 4
Feature sets.

Feature set	No.	Feature attribute	New feature sets	Expression
DNS-based	1	Internal source host ID	*Int_Src*	$H_{int-src}$
	2	Resolved external host ID	*Ext_Dst*	$H_{ext-dst}$
Host ID-based	3	Internal destination host ID	*Int_Dst*	$H_{int-dst}$
Network flow-based	4	Number of bytes (Data count)	*Byte_Count*	$\sum byt$
	5	Number of connections (Connection count)	*Conn_Count*	$\sum con$
	6	Number of external host destinations	*Ext_Dst-Count*	$\sum H_{ext-dst}$
Time value-based	7	Time window of activity (Duration)	*T_window*	Δt
Protocol	8	The protocol used in the communication	*Prot*	p

Table 5
The Red Team dataset.

SN	Time	User ID	Source host ID	Destination host ID
1	150 885	U620@DOM1	C17693	C1003
2	830 550	U1653@DOM1	C22409	C754
3	2 557 047	U737@DOM1	C19932	C108
N	….	….	….	…

activities which include data exfiltration and sabotage which is the last phase of the attack process (*cf.* Fig. 1). As such, there exists an underlying APT attack network (sub-graph) within the general communication network.

5.1. State changes of nodes in the APT-AN

In order to depict the various states through which APT-AN nodes transition, a finite state machine is constructed. The finite state machine is based on the three stages of the proposed APT attack lifecycle in Fig. 3 and the time-dependent multi-slice network in Fig. 2. The resultant finite state machine is illustrated in Fig. 6.

Four security states $\boldsymbol{S_n}$ are defined as:

- $\boldsymbol{S_0}$. – the secured state of the system before APT intrusion (secure)
- $\boldsymbol{S_1}$. – the state of the system after APT malware infection but before traversal (susceptible-propagating)
- $\boldsymbol{S_2}$. – The state of the system when infected nodes are used for lateral traversal and privilege escalation (active-propagating)
- $\boldsymbol{S_3}$. – The state of the system when the target node is actively communicating with the C2 to exfiltrate data (exfiltration or sabotage)

As such, the transitions of the nodes in the APT lifecycle can be summarized as: $\boldsymbol{S_0}$\{secure\}$\longrightarrow$ $\boldsymbol{S_1}$\{susceptible-propagation\}$\longrightarrow$ $\boldsymbol{S_2}$\{active-propagation\}$\longrightarrow$ $\boldsymbol{S_3}$\{exfiltration/sabotage\}.

Nodes in state $\boldsymbol{S_0}$ represent the secure state prior to any breach. Nodes in state $\boldsymbol{S_1}$ present susceptible nodes that have been identified and infected by the attacker. Nodes in state $\boldsymbol{S_2}$ are those that have been successfully exploited by the attacker and they are used as pivot nodes to accomplish lateral traversal and privilege escalation. Nodes in state $\boldsymbol{S_3}$ are the target nodes which are actively exfiltrating data or where the data is sabotaged in the case of a DoR attack. Since a node in the APT-AN can be in any of the defined states, binary variables can be used for denotation. Let the binary variable x denote the susceptible-propagation activity, y to denote the active-propagation activity, and z to denote the exfiltration or sabotage. Therefore, at any given point in the APT attack lifecycle, the state of a node can be given as a binary function $f\{\boldsymbol{S_n}(x,y,z)\}$ *where* $x,y,z \in \mathbb{N}_2$. Considering that $x,y,z \in \{0,1\}$, it follows that these binary variables have complements denoting the absence of the corresponding attack activity. These complements are denoted as $\bar{x},\bar{y}$ *and* $\bar{z}$ respectively. Clearly, the state function representative of the susceptible-propagation phase $\boldsymbol{S_1}$ is expressed as:

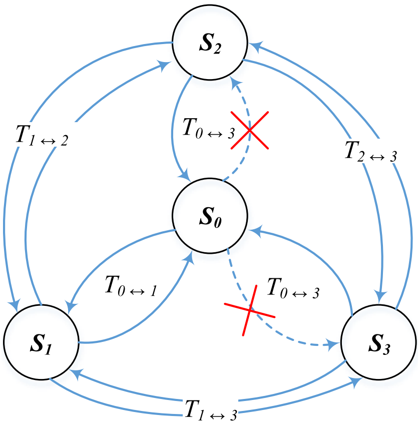


Fig. 6. State transitions of the nodes participating in the APT-AN.

$$f\{\boldsymbol{S_1}(x,y,z)\} = x\cdot\bar{y}\cdot\bar{z}, \qquad where\ x,y,z \in \mathbb{N}_2 \tag{10}$$

Eq. (10) implies that nodes in state $\boldsymbol{S_1}$ can only be infected and carry out reconnaissance but not laterally traverse nor exfiltrate data. Using K-maps, the state function of the active-propagation representative of the state $\boldsymbol{S_2}$ is derived as:

$$f\{\boldsymbol{S_2}(x,y,z)\} = x\cdot y\cdot\bar{z} + \bar{x}\cdot y\cdot\bar{z}, \qquad where\ x,y,z \in \mathbb{N}_2 \tag{11}$$

Eq. (11) implies that nodes in state $\boldsymbol{S_2}$ are laterally traversing but can also be carrying out reconnaissance. The term $x\cdot y\cdot\bar{z}$ implies that the same infected node from $\boldsymbol{S_1}$ is used in lateral traversal where $\bar{x}\cdot y\cdot\bar{z}$ implies that a newly exploited node explicit to the active-propagation phase is used for lateral traversal. Using K-maps, the state function of the exfiltration/sabotage representative of the state $\boldsymbol{S_3}$ is derived as:

$$f\{\boldsymbol{S_3}(x,y,z)\} = x\cdot y\cdot z+\bar{x}\cdot y\cdot z+\bar{x}\cdot\bar{y}\cdot z, \qquad where\, x,y,z\in\mathbb{N}_2 \quad (12)$$

Eq. (12) implies that nodes in state $\boldsymbol{S_3}$ are exfiltrating, data but can also be used for lateral traversal and or reconnaissance if they are from the state $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ respectively. The term $x \cdot y \cdot z$ implies that the same node exploited both in state $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ is used for data exfiltration. The term $\bar{x} \cdot y \cdot z$ implies that the node exploited only in the state $\boldsymbol{S_2}$ is used for data exfiltration. Finally, the term $\bar{x}\cdot\bar{y}\cdot z$ implies that a newly exploited node exclusive to the state $\boldsymbol{S_3}$ is used for data exfiltration. The proposed framework uses network statistics to detect suspicious hosts in these 3 states, which correspond to the 3 stages of the APT attack lifecycle.

In order to detect suspicious APT activities in the three phases of the attack lifecycle, a specific set of features from Table 5 are applied to each of the S_n stages. Features that characterize the properties of a dynamic complex network are thus used. To this effect, those features that reflect the vertex degrees and the clustering coefficients, and other network flow statistics are used. Table 6 summarizes the assignment of FSM states to various clusters using properties of complex networks.

5.2. Overview of the proposed APT detection framework

Fig. 7 shows an overview of the proposed framework and the main components thereof. The input data from the dataset is partitioned into network flows and DNS logs from which features that are used in the detection process are drawn. The extracted features are then normalized for uniformity of analysis and the ranking computation is used to produce an output of a ranked list of hosts suspected of participating in APT activities. Since the LANL dataset is too large and in order to perform efficient analysis, data reduction techniques are used and thus reduce the volume of the data while preserving the communication structure between source and destination hosts. Null entries that have question marks "?" in the value portion of the column are removed, and equally repetitive and redundant entries are removed. The Authentication and Processes information and the associated User Counts are also removed. When dealing with DNS communications, all other port numbers are dropped but retain port 53 for DNS under User Datagram Protocol (UDP) and port 5353 Transmission Control Protocol (TCP). To achieve this, two databases were created under the MS SQL server; one database for DNS communications and the other for generic network flow traffic. MS SQL queries were used to efficiently reduce the volume of the data.

For the DNS dataset, attention is concentrated on A records which in turn significantly reduces the average volumes of DNS requests per day. Furthermore, the network traffic is partitioned into internal and external communications. The internal communication traffic is used analysis of the lateral movement APT phase whilst the external communication traffic is used in the analysis of the infiltration and exfiltration APT phases. The time frame of the data is partitioned to days, weeks and months. A network-based approach is adopted because network traffic flows can be acquired and analyzed easily than logs of individual hosts in a large-scale enterprise network [54]. The computation and ranking of suspicious hosts are elaborated in the subsequent sections. The goal is not to pinpoint which host is participating in APT activities but rather identify a list of potentially compromised hosts participating in APT activities. This is particularly important to security analysts who would otherwise need to skim through huge volumes of network traffic logs for weak signals of APT activities which is not a trivial task.

5.3. The semi-supervised learning approach

This section presents the semi-supervised learning approach to APT detection that takes advantage of the huge amount of unclassified data from the Los Alamos dataset to perform classification of suspicious hosts participating APT activities, using few labeled instances from the Red Team. Fig. 8 shows the proposed semi-supervised learning approach.

The proposed semi-supervised approach shown in Fig. 8 shows that in addition to unlabeled raw data (network flow traffic and DNS dataset), a set of Red Team labeled data with features depicted in Table 5 is also available. The semi-supervised approach uses features derived from properties of complex networks of the unlabeled data (clustering coefficients and vertex degrees) to create a supervised model. The feature extraction step in the supervised section of the proposed approach uses a mapping scheme to extract hosts from the unlabeled dataset. The mapping scheme uses the source host ID, destination host ID, and time in order to locate a host and generate APT features from the dataset.

As such, a semi-supervised learning approach is proposed where clusters of different sizes are derived mainly based on the clustering coefficient and vertex degree. To analyze the normalized data and derive a ranked list of suspicious hosts participating in APT related activities, an enhanced semi-supervised algorithm based on the Shared Nearest Neighbour (SNN) clustering algorithm [55] is used. The SNN clustering defines similarity or proximity between two nodes in terms of the number of directly connected neighbors they have in common. This suits its applicability in complex networks since the clustering coefficient and vertex degrees are dictated by neighbor relations.

As such, the SNN algorithm is adopted which apart from considering direct associations between nodes also considers indirect connections. This provides for an ability to detect similarities between nodes that are not necessarily adjacent. Additionally, SNN has the ability to handle clusters of varying sizes, densities and shapes. As such, two nodes that are relatively close but belong to different clusters are handled effectively. The 5-step high-level overview of the enhanced SNN algorithm is shown in Fig. 9. The native SNN algorithm is extended based on KNN properties by classifying the resultant clusters using the clustering coefficient and vertex degree.

In step 1, the adjacency matrix is derived from the graph depicting network communications while the proximity matrix is computed in step 2. The proximity matrix is used to determine the extent to which the two given nodes in a graph are connected or belong to a cluster or common group. Whereas the connectivity (adjacency) matrix is used to determine the linkability of any set of nodes, the proximity matrix is used to determine which pair of vertices share the same neighbors. Since the Shared Nearest Neighbor defines similarity or proximity between any given nodes in terms of the number of neighbors they have in common, it is this definition of proximity measure that forms the basis of the SNN algorithm. Semantics on the derivations of the matrices are elaborated . Using these matrices, the corresponding KNN graph is constructed in step 3. In step 4, the SNN graph is generated from the resultant KNN. Finally, step 5 derives clusters from the SNN graph based on different network statistics discussed earlier and classify the hosts in each cluster to get a list of ranked suspicious hosts. Algorithm 1 illustrates the enhanced SNN algorithm.

Since there are three nested loops in the algorithm, the resultant computational complexity is of the order $O(n^3)$. Since only the k nearest neighbors need to be stored, the resultant space complexity is of the order $O(k \cdot n)$. Even though it evident that the computational complexity thus far is not less than that of the generic SNN algorithm [56], there are a number of optimization

Table 6
Relationship between the FSM states and the clusters.

Cluster	Byte_Count ($\sum byt$)	Conn_Count ($\sum con$)	Int_Dst-Count ($\sum H_{int-dst}$)	Ext_Dst-Count ($\sum H_{ext-dst}$)	Vertex degree (ΔK_i)		Cluster coeff. (ΔC_i)		State
					(ΔK_i^{int})	(ΔK_i^{ext})	(ΔC_i^{int})	(ΔC_i^{ext})	
C_0	*very high*	*high*	*Insignif.*	*Very high*	*low*	*very high*	*low*	*high*	$\boldsymbol{S}_3$
C_1	*var*	*var*	*var*	*var*	*Insignif.*	*Insignif.*	*Insignif.*	*Insignif.*	$\boldsymbol{S}_0$
C_2	*avg*	*high*	*very high*	*low*	*very high*	*low*	*high*	*low*	$\boldsymbol{S}_2$
C_3	*var*	*var*	*var*	*var*	*Insignif.*	*Insignif.*	*Insignif.*	*Insignif.*	$\boldsymbol{S}_0$
C_4	*avg*	*high*	*low*	*high*	*very low*	*high*	*very low*	*high*	$\boldsymbol{S}_1$
C_i	....	....	....	....	....	....	....	....	$\boldsymbol{S}_{[0,3]}$

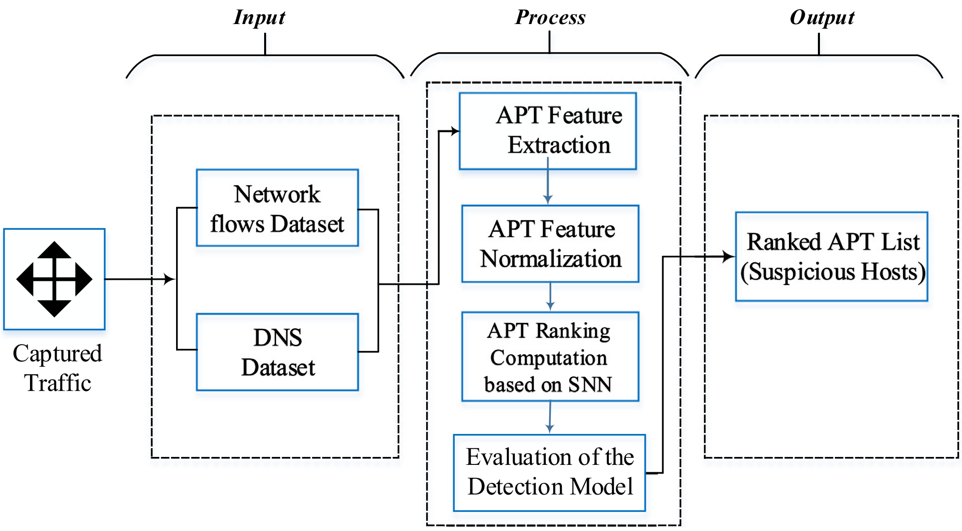

Input
Process
Output
Captured Traffic
Network flows Dataset
DNS Dataset
APT Feature Extraction
APT Feature Normalization
APT Ranking Computation based on SNN
Evaluation of the Detection Model
Ranked APT List (Suspicious Hosts)

Fig. 7. Overview of the proposed APT detection framework.

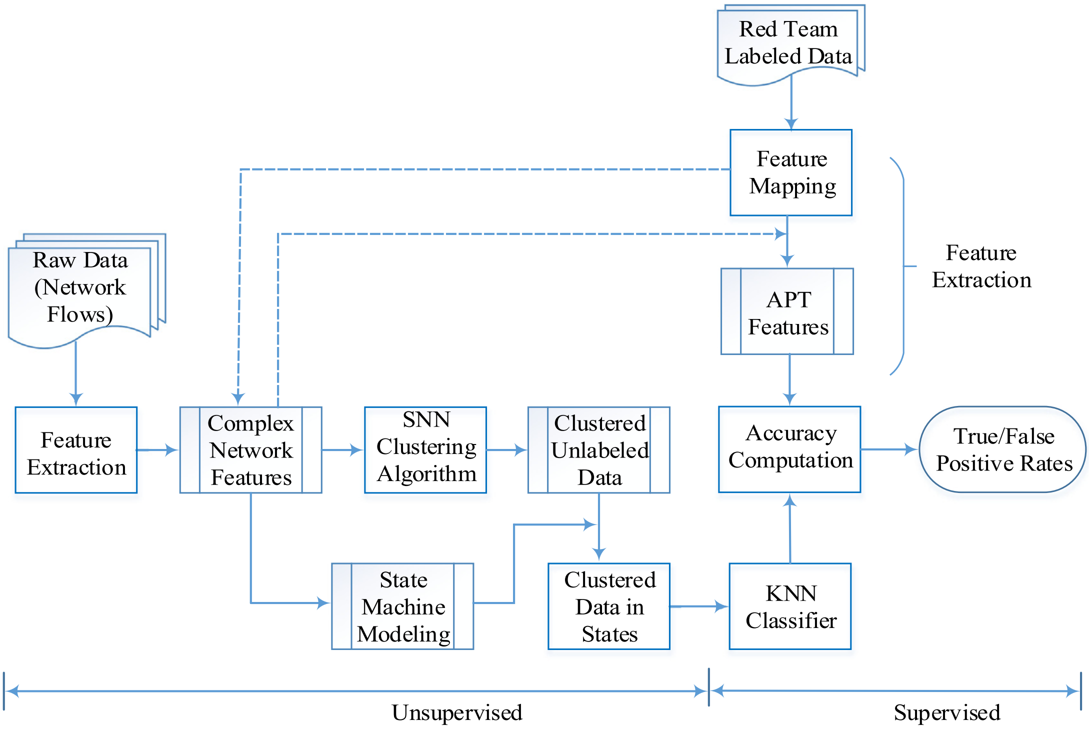


Fig. 8. A semi-supervised learning approach for APT attack detection.

techniques [57] to reduce this relatively high computational overhead. The basic idea in these optimization techniques is to find an economical way of computing the nearest neighbors of a vertex by restricting the considered number of nodes. Optimization of the enhanced algorithm is committed to future works.

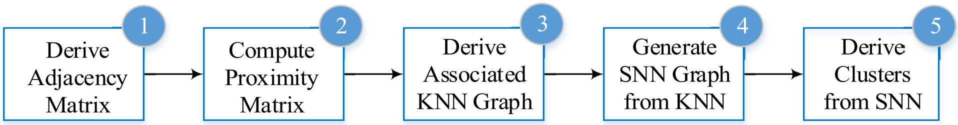


Fig. 9. A high-level overview of the enhanced SNN algorithm.

```
Algorithm 1: Enhanced SNN for detecting suspicious APT activities
Input: $G$ - an undirected graph and $k$ number of shared nearest neighbors
Output: $\boldsymbol{L}^*$ - ranked list of the suspicious host participating in APT activities.

Initialize $\boldsymbol{G}^*$ with $|V(G)|$ vertices, no edges
foreach $i = 1$ to $V(G)$ do
    foreach $j = i + 1$ to $V(G)$ do
        $counter = 0$
        foreach $m = 1$ to $V(G)$ do
            if vertex $i$ and vertex $j$ both have an edge with vertex $m$ then
                $counter = counter + 1$
            end
        if $counter \geq k$ then
            Connect an edge between vertex $i$ and vertex $j$ in $\boldsymbol{G}^*$
            for $\boldsymbol{S} \leftarrow 0$
                if $\triangle K_i > 1$ at time $t^*$ for external communications
                    then $(H_{int-src}) \in \boldsymbol{S_1}$
                else if $\triangle K_{\boldsymbol{i}}'' > \triangle K_i$ && $\triangle C_i > 1$ && $\triangle C_i > [\triangle C_{i-1,\dots 0}]$ then
                    then $(H_{int-src}) \in \boldsymbol{S_2}$
                else if $\triangle K_{\boldsymbol{i}}'' > 1$ && $M_v > (X^*_{threshold})$ for time window $\triangle t$
                    then $(H_{int-src}) \in \boldsymbol{S_3}$
                end if
        end
end
return $\boldsymbol{L}^*$
```

As such, the proposed semi-supervised approach consists of two phases: an unsupervised phase that produces features from properties of complex networks based on vertex degrees and clustering coefficients. This phase uses finite state machine (FSM) modeling to assign states to clusters, and a supervised phase that learns and trains the model. This phase uses the KNN classifier and the Red Team data. In short, the proposed semi-supervised learning approach uses the unsupervised learning method to extract features from the unlabeled dataset and the supervised model classifies this data into APT attack stages using features from properties of complex networks. The unsupervised phase utilizes the shared nearest neighbor clustering whilst the supervised phase utilizes the KNN. The semi-supervised learning approach is summarized in Algorithm 2.

The unlabeled and labeled data $\{X^i_{u-d}, X^i_{l-d}\}$ from network flows and the Red Team respectively are initialized and read in step 1. Step 2 normalizes the data by converting the input values to a common scale $[\overline{X^\imath_{u-d}}, \overline{X^\imath_{l-d}}]$. This enables us to make an effective comparison of the variations of the clustering coefficient and vertex degree. The process of normalization is elaborated in Section 5.5. The properties of complex networks $k\ and\ c$ (vertex degree and clustering coefficient respectively) are extracted from the unlabeled data $\{X^i_{u-d}\}$ in step 3. The labeled data from the Red Team, with the help of features from Table 7, is used in a mapping scheme in step 4 to locate a host and generate APT features from the dataset. The SNN unsupervised clustering algorithm is used to create clusters in step 5 via the process $\{X^i_{u-d}\} \rightarrow SNN \Rightarrow C_0, C_1, C_2, \dots, i$. Using the methodology explained in Table 1, step 6 employs FSM modeling $C_0, C_1, C_2, \dots, i \rightarrow FSM(S_n) \Rightarrow C_i^{S_n}$ to assign states to the clusters $C_0, C_1, C_2, \dots, i$ generated in step 5. The KNN supervised algorithm is applied to the datasets $\{\overline{X^\imath_{l-d}}\}$ and $\{\overline{X^\imath_{u-d}}\}$ in step 7. The clustered data with states $\{C_i^{S_n}\}$ is classified by the KNN algorithm in step 8. The True Positive and False Positive rates (TP && FP) are computed in step 9 and the corresponding TP && FP rates for the clustered states are returned in step 10.

5.4. APT activity detection in the 3 stages

This section presents the semantics of APT activities detection in the 3 stages from the proposed ATP attack lifecycle. As such, detection methodologies in the Infiltration, Lateral Movement, and C2 Beaconing and Data Exfiltration phases are presented here.

5.4.1. Detection in the infiltration phase

At infection, the attacker deploys APT malware to the victim but the victim first beacons to the attacker's C2 servers to acquire RATs. Therefore, this stage seeks to detect initial communications to these remote C2 servers which are essentially new and rare destinations. To this end, the initial DNS requests and the subsequent communications between internal and external hosts are analyzed and likewise evaluation of the flows between them. To this effect, graphs are built where the vertices represent internal

```
Algorithm 2: Semi-supervised learning approach for APT attack detection
Input: $X^i_{u-d} = \{x^1_{l-d}, x^2_{l-d}, x^3_{l-d}, \ldots, x^i_{l-d}\}$, unlabeled network flows data
            where $x^i_{l-d} \in R^n, i = 1,2,3,\ldots,n$
       : $X^i_{l-d} = \{x^1_{u-d}, x^2_{u-d}, x^3_{u-d}, \ldots, x^i_{u-d}\}$, labeled Red Team data
            where $x^i_{u-d} \in R^n, i = 1,2,3,\ldots,n$
Output: $TP$ && $FP$ rates - APT attack detection accuracy
1. Read the unlabeled & labeled network flow dataset
2. Normalize original data $X^i_{l-d}, X^i_{u-d}$, get the normalized data $\overline{X^i_{u-d}}, \overline{X^i_{l-d}}$
3. Extract $k$ and $c$, complex network features from $X^i_{u-d}$
4. From $X^i_{l-d}$, map corresponding $k$ and $c$ in $X^i_{u-d}$, generate APT features
5. Cluster $X^i_{u-d} \rightarrow SNN \Rightarrow C_0, C_1, C_2, \ldots, i$
6. Generate cluster states $C_0, C_1, C_2, \ldots, i \rightarrow FSM\ (S_n) \Rightarrow C_i^{S_n}$
7. Train supervised KNN with $\overline{X^i_{l-d}}$ and $\overline{X^i_{u-d}}$
8. Classify $C_i^{S_n}$ with the KNN
9. Compute APT detection accuracy $TP$ && $FP$ rates based on SNN & KNN
10. Return $TP$ && $FP$ rates for $C_i^{S_n}$ clusters
```

and external hosts while the weighted edges correspond to the number of flows with the external host for a specified DNS request of a new and rare destination. This corresponds to $\boldsymbol{S_1}$, the first stage of the APT attack lifecycle. Let $\sum H_{ext-dst}(t)$ be a set of external destination hosts contacted up to time t and $\sum H_{ext-dst}(t+1)$ be a set of external destination hosts contacted by time $t+1$. As such, ΔK_i, a change in the vertex degree, depicting a change in the number (behavior) of communications to external hosts is thus computed as:

$$\Delta K_i = \frac{k_t}{k_{t+1}} = \frac{\sum H_{ext-dst}(t)}{\sum H_{ext-dst}(t+1)} \tag{13}$$

It is worth noting that if the hosts contacted for Δt are identical, then $\Delta K_i = 1$ implying no change in the vertex degree and communication pattern. A value of $\Delta K_i > 1$ implies an increase in the vertex degree denoting implying the given host H_i has contacted new external hosts which can harbor a potential C2. Δt is set to 24 h which is a reasonable time period for security analysts to evaluate daily traffic logs. In other instances, the value of Δt is adjusted to suit the need.

5.4.2. Detection in the lateral movement phase

Detection of suspicious APT activities in the lateral movement phase is carried out by considering the variations in the vertex degree k and the clustering coefficient C of an internal node with respect to internal communications. At this point, only communications to hosts within the network are considered because the attacker seeks to discover more vulnerable hosts, increase his knowledge about the targets and escalate privileges whenever necessary. This corresponds to $\boldsymbol{S_2}$, the second stage of the APT attack lifecycle. As such, two dynamic complex network parametric values ΔC_i and $\Delta K_i'$ are defined to denote the growth of the APT-AN with respect to the clustering coefficient and vertex degree respectively. $\Delta K_i'$ is computed according to Eq. (13), the only difference is that the connections are with respect to internal host destinations. As the vertex degree of the suspicious host grows, so does the associated clustering coefficient since connections to new internal hosts create new clusters. The clustering coefficient parametric value that depicts this change is computed as:

$$\Delta C_i = \frac{C_t}{C_{t+1}} = \frac{\sum H_{int-dst}(t)}{\sum H_{int-dst}(t+1)} \tag{14}$$

where C_t is the clustering coefficient up to time t and C_{t+1} is the new clustering coefficient after time $t+1$. Likewise, the time window Δt for the variation in clustering coefficient in the interval between t and $t+1$ follows that of ΔK_i for each given host.

5.4.3. Detection in the C2 beaconing and exfiltration phase

To detect suspicious hosts that might be involved APT data exfiltration to the C2 centers at time t, the following are considered:

(1) $\Delta K_i''$ - the new vertex degree in stage $\boldsymbol{S_3}$, denoting the number of external hosts destinations communicated with, denoted as x_t^1
(2) $\sum byt$ - the amount of data uploaded to the external destinations denoted as x_t^2
(3) $\sum con$ - the number of connections to the external hosts denoted as x_t^3

In order to detect which internal hosts ($H^i_{int-src}$) might be involved in APT data exfiltrations, attention is focused on data sent from the enterprise network to external destination hosts. This corresponds to $\boldsymbol{S_3}$, the third stage of the APT attack lifecycle. To this end, a feature vector X is formulated for each internal host that is uploading data out of the network as $X_t = (x_t^1, x_t^2, x_t^3)$. This parameter characterizes the initial position of a host and any traffic variations due to suspicious APT activities cause a deviation from the centroid. To estimate the variation of this behavior, the movement vector as a Euclidean difference in the feature space [58,59] is adopted. This parameter M_v is expressed as:

$$M_v = \| \Delta_k^c \| = \sqrt{\sum_{i=1}^{N} \left(\frac{x_t^i - X_t^*\,(\Delta t)}{X_t^*\,(\Delta t)} \right)^2} \tag{15}$$

where the outgoing traffic statistics Δ_k^c quantifies changes in the clustering coefficient ΔC_i and vertex degree ΔK_i, where Δt is the time window, and X_t^* denotes the centroid of the feature space.

5.5. Feature normalization

The current dataset presents high volumes of the traffic that was repeatedly generated at short time intervals. Such data tends

Table 7
Five clusters with complex networks characteristics.

Attribute	Full data	Cluster ID				
		C0	*C1*	*C2*	*C3*	*C4*
Byte_Count	0.7	0.971	0.761	0.984	0.860	0.015
Conn_Count	0.62	0.793	0.761	0.837	0.581	0.997
Ext_Dst-Count	0.45	0.659	0.372	0.751	0.861	0.330
Int_Dst-Count	0.52	0.470	0.936	0.907	0.742	0.019
Packet_Count	0.46	0.985	0.733	0.97	0.792	0.091
Protocol	0.56	0.549	0.523	0.8	0.578	0.979
Int-Conn_Count	0.67	0.472	0.831	0.989	0.7	0.510
T_window	0.38	0.34	0.525	0.87	0.937	0.30

to be correlated and this causes measurement bias due to the heavy-tailed distribution characteristics depicted in Fig. 5. In order to make an effective comparison of the variations of the clustering coefficient and vertex degree, the data is normalized by converting the input values to a common scale. The goal of this normalization is to create a common scale without distorting any differences that may exist between the ranges of the values and thus minimize loss of information. This is important because it provides for data consistency and minimization of redundancy. As such, the normalization stage of the proposed framework shown in Fig. 7 accomplishes this task. The quartile weighted mean [60, 61] is adopted for normalization as opposed to the common min–max scaling [62] because the latter yields poor results of values close to zero due to the heavy-tailed distribution characteristics of the data. The quartile weighted mean (QWM) is calculated as:

$$QWM\,(x) = \frac{Q_{1/4}\,(x) + 2\cdot Q_{1/2}\,(x) + Q_{3/4}\,(x)}{4} \tag{16}$$

where $Q_i(x)$ is the *i-th* quantile of the values in the dataset. In this way, the data can be effectively compared to depict variations in the clustering coefficients and vertex degrees.

6. Results and discussion

Application of the SNN clustering process to the dataset yields the results shown in Table 6. This yields 5 clusters namely; C0, C1, C2, C3, and C4. Cluster C0 has a high *Byte_Count* (97.1%) and a high *Conn_Count* (79.3%). It also has a high *Ext_Dst-Count* (65.9%) compared to *Int_Dst-Count (47%)*. This implies that hosts in this cluster have a higher vertex degree and clustering coefficient with regards to external communications. Furthermore, this cluster has a relatively small time window. These are typical characteristics of hosts in $\boldsymbol{S_3}$ of the APT lifecycle.

On the contrary, cluster C2 has *Byte_Count (98.4%)* and *Conn_Count (83.7%)* but the *Int_Dst-Count (90.7%)* is greater than *Ext_Dst-Count (75.1%)*. This implies that hosts in this cluster have a higher clustering coefficient and vertex degree with regards to internal communications. The time window for activities in this cluster is relatively higher than C0. These are typical characteristics of hosts in $\boldsymbol{S_2}$ of the APT lifecycle.

A lower *Byte_Count (1.5%)* and a high *Conn_Count (99.7%)* corresponding to *Ext_Dst-Count (33%)* instead of *Int_Dst-Count (19%)* for a smaller time window in cluster C4 entail that hosts in this cluster communicate more with external hosts. Furthermore, hosts in this cluster have a high *Protocol (97.9%)* value implying the use of a common protocol, such as DNS, in the early stages of communication. These characteristics correspond to hosts in $\boldsymbol{S_1}$ of the APT lifecycle.

The clusters C1 and C3 have relatively average network statistics that depict the behavior of benign hosts. The high *Int_Dst-Count (93.6%)* in C1 corresponds to a high *Int-Conn_Count (83.1%)* which is a correlation expected of normal network traffic. Equally in cluster C3, *Byte_Count (86%)* corresponding to *Ext_Dst-Count (86.1%)* which is supplemented by average values of other characteristics in the same range. The variations in the clustering coefficient (ΔC_i) and vertex degree (ΔK_i and $\Delta K_i''$) in these network traffic statistics in the respective clusters depict the overall movement of the movement vector (M_v) from the feature centroid (X_t^*). Table 7 shows a summary of the dynamic characteristics of the clusters in relation to the 3 APT attack lifecycle stages.

After generating the clusters and associating them with the APT attack phases, the Red Team data (labeled dataset) is used for classification and also used to evaluate the effectiveness of the proposed approach. This is because the hosts in the Red Team data are technically labeled as malicious for exhibiting APT attack activities. However, they do not show which phase of the APT attack process they belong to. To address this challenge, the associated vertex degrees and clustering coefficients are mapped to the corresponding APT attack phase (see Table 8).

Since an SNN graph is a special type [63] of a KNN graph where an arc exists between two nodes $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$ if there are a minimum of $\boldsymbol{k}$ nodes adjacent to both $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$, the KNN algorithm is adopted for classification in the supervised learning portion of the proposed approach. With the help of the Red Team, the classification algorithm is run using the dataset labeled.

Considering the fact that the proposed approach detects suspicious APT activities in three different stages, three distinct Receiver Operator Characteristic (ROC) curves are generated representative of the 3 stages. Since spline interpolation incurs less error than linear interpolation [64], and the interpolant is smoother, the use B-spline interpolation for the output smooth curve is adopted.

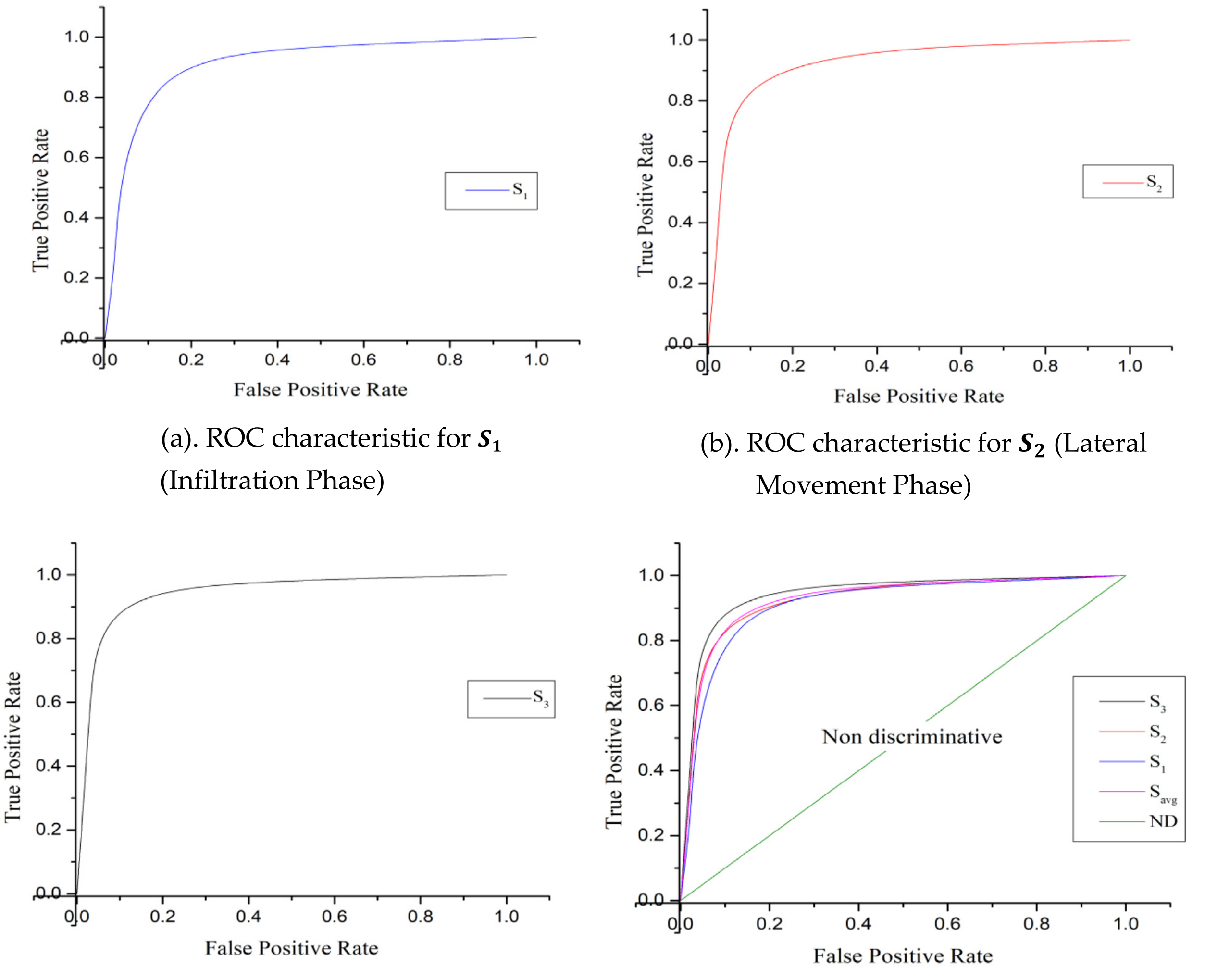
The ROC curve for $\boldsymbol{S_1}$ is shown in Fig. 10(a). Even though splines allow the accurate modeling of more general classes of geometry, it is acknowledged that a very large smoothing factor results in underfit whereas a very small smoothing factor will result in too much noise (overfit). Non-negative parameter (smoothing factor) are used to specify the smoothness of the interpolated curve in the spline interpolation. This factor helps to control the balance between smoothing and closeness.

The ROC curve for $\boldsymbol{S_2}$ is shown in Fig. 10(b) and the ROC curve for $\boldsymbol{S_3}$ is shown in Fig. 10(c). The average of the 3 curves is evaluated and consolidated into a single graph depicting their respective areas under the curve relative to the average and the non-discriminative (ND) characteristics. The ND characterizes the expectation for random guessing (classification). As such, the closer to ND a ROC curve is, the poorer the performance. The resultant characteristics of the ROC curves are shown in Fig. 10(d).

It is evident from the ROC curve in Fig. 10(c) that the classification of APT attack detection in the data exfiltration phase ($\boldsymbol{S}_3$) has the highest accuracy while the infiltration phase ($\boldsymbol{S_1}$) has the lowest accuracy. The high accuracy in stage $\boldsymbol{S_3}$ is largely attributed to the abundant features exhibited in this stage. Apart from the variations in the clustering coefficients and vertex degrees that are used in stages $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$, detection in stage $\boldsymbol{S}_3$ encompasses *Byte_Count* and *Conn_Count*. These features are used to detect anomalies in that if the number of external destination hosts ($H_{ext-dst}$) contacted within a given time frame by an internal host ($H_{int-src}$) stays the same while the outgoing Byte_Count or the Conn_Count greatly increases, it may correspond to data exfiltration activities which are a feature of APT attacks.

In the same manner, if the number of flows generated by an internal host ($H_{int-src}$) increases greatly in a given time window, the number of external destinations ($H_{ext-dst}$) is expected to increase as well. As such, this stage of the attack process has additional detection features, hence the higher accuracy when compared to the other two stages, $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$.

The classification accuracy of the average ROC curve of the three phases ($\boldsymbol{S}_{avg}$) lies above those of $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$. It is worth noting



(a). ROC characteristic for S_1 (Infiltration Phase)

(b). ROC characteristic for S_2 (Lateral Movement Phase)

(c). ROC characteristic for S_2 (Exfiltration Phase)

(d). ROC characteristics of the classification model

Fig. 10.

Table 8
Correlation between the clusters and the 3 APT attack stages.

Cluster	APT attack phase	State	Vertex degree (ΔK_i)		Clustering coefficient (ΔC_i)	
			Internal (ΔK_i^{int})	External (ΔK_i^{ext})	Internal (ΔC_i^{int})	External (ΔC_i^{ext})
C0	Exfiltration	S_3	*low*	*Very high*	*low*	*high*
C1	–	S_0	–	–	–	–
C2	Lateral Movement	S_2	*Very high*	*low*	*high*	*low*
C3	–	S_0	–	–	–	-
C4	Infiltration	S_1	*Very low*	*high*	*Very low*	*high*

that although the accuracy of S_1 is greater than that of S_{avg} up to the decision threshold TP rate = 0.81, the difference between the AUC (Area under the ROC Curve) under the S_{avg} ROC curve and S_2 ROC curve between FP rate = 0.79 and FP rate 1.00 is way greater than between FP rate = 0 and FP rate = 0.79. As such, the accuracy of S_{avg} ROC curve is higher than that of S_2. In view of this optimization efforts should seek to increase the accuracy of S_1 and S_2 beyond S_{avg} and near to S_3.

The detailed characteristics of the model for the hosts classified using the clusters are shown in Table 9.

The model has a relative absolute error of 18.1% and a root-relative squared error of 47.2%. Correctly classified instances represent 90.4% while incorrectly classified instances represent 9.62%. The model has good performance because the weighted average of the ROC Area is near 1 and way above the non-discriminative characteristic (N.D) which represents equal TP and FP rates.

The ROC Area [65] entails the predictive characteristics of the model to distinguish between the true positives and the true negatives. As such, the model does not only predict a positive value as a positive but as well as a negative value as a negative. The TP Rate represents the instances that are correctly classified as a given class which essentially is the rate of true positives. The FP Rate represents which of the instances falsely classified as a given class which essentially is the rate of false positives.

The PRC, as opposed to the ROC Area, represents the behavioral characteristics of Precision Vs Recall. The Precision value denotes the ratio of instances that are true of a given class divided

Table 9
The characteristics of the model.

Class	TP Rate	ROC Area	FP Rate	PRC Area	Precision	Recall	F-Measure	MCC
$\boldsymbol{S_3}$	0.938	0.972	0.056	0.882	0.882	0.938	0.909	0.913
$\boldsymbol{S_2}$	0.882	0.971	0.063	0.941	0.938	0.882	0.909	0.913
$\boldsymbol{S_1}$	0.895	0.891	0.105	0.840	0.895	0.895	0.895	0.832
Weig. Av.	0.905	0.945	0.075	0.888	0.905	0.905	0.904	0.886

Table 10
Top-12 hosts with suspicious APT activities.

Host	Byte_Count	Dst_Count	DNS_Count	Stage ($\boldsymbol{S_n}$)	Cluster
$H^1_{int-src}$	6642654468	6365	687	$\boldsymbol{S_3}$	*C0*
$H^2_{int-src}$	5462733941	7548	570	$\boldsymbol{S_3}$	*C0*
$H^3_{int-src}$	3188938177	4781	48	$\boldsymbol{S_3}$	*C0*
$H^4_{int-src}$	2798636451	1079	59	$\boldsymbol{S_3}$	*C0*
$H^5_{int-src}$	2059864734	1207	12	$\boldsymbol{S_2}$	*C2*
$H^6_{int-src}$	1300199349	793	15	$\boldsymbol{S_2}$	*C2*
$H^7_{int-src}$	1078447453	19	11	$\boldsymbol{S_2}$	*C2*
$H^8_{int-src}$	899475045	89	7	$\boldsymbol{S_2}$	*C2*
$H^9_{int-src}$	630743972	5740	1742	$\boldsymbol{S_1}$	*C4*
$H^{10}_{int-src}$	443783383	363	706	$\boldsymbol{S_1}$	*C4*
$H^{11}_{int-src}$	200432761	5858	929	$\boldsymbol{S_1}$	*C4*
$H^{12}_{int-src}$	98437877	657	100	$\boldsymbol{S_1}$	*C4*

by the sum of instances classified as that given class. The Recall value denotes the ratio of instances classified as a class divided by the actual sum in that given class. As such, this is equivalent to the TP rate. The F-Measure is a combined measure that depicts the ratio of double the product of Precision and Recall divided by the sum thereof, i.e. $\frac{2\cdot Precision*Recall}{\sum(Precision+Recall)}$. The MCC is the measure of the quality of binary classifications taking into account true and false positives and negatives. It is a balanced measure that has a range $[-1, 1]$, with -1 denoting a completely wrong classifier and 1 indicating the opposite. The variations in the clustering coefficient (ΔC_i) and vertex degree (ΔK_i and $\Delta K_i''$) in these network traffic statistics in the respective clusters depict the overall movement of the movement vector (M_v) from the feature centroid (X_t^*).

Using a threshold value, a list of suspicious hosts with high values of $\parallel \Delta_k^c \parallel$ is generated. Since the ultimate goal of almost all APT activities is data exfiltration, a higher weighting is given to hosts depicting characteristics of the $\boldsymbol{S_3}$ the phase of the APT lifecycle. Table 10 shows a ranked list of suspicious hosts generated from the 4 clusters discussed above. An average of 4 hosts for each stage of the APT lifecycle are selected. The ratio of the *Byte_Count* and the *Dst_Count* depicts the average weight of the edges from source to destination whereas the Dst_Count depicts the average vertex degree for that given time window.

It is worth noting that the destination addresses in stage 2 ($\boldsymbol{S}_2$) refer to internal destinations within the network ($H_{int-dst}$) whereas those in $\boldsymbol{S_3}$ and $\boldsymbol{S_1}$ refer to external addresses ($H_{ext-dst}$). Evidently, the *DNS_Count* in $\boldsymbol{S_2}$ is less than in $\boldsymbol{S_3}$ and $\boldsymbol{S_1}$ because $\boldsymbol{S_2}$ is representative of the *lateral movement* phase of the APT attack-cycle which is characterized by few DNS requests due to local network host resolutions. The set of hosts $\{H^1_{int-src}, H^2_{int-src}, H^3_{int-src}, H^4_{int-src}\}$ in stage $\boldsymbol{S_3}$ exfiltrate more data and have a relatively high variation of the vertex degree $\Delta K_i''$ which is a characteristic of this stage of the APT attack cycle. The set of hosts $\{H^5_{int-src}, H^6_{int-src}, H^7_{int-src}, H^8_{int-src}\}$ in stage $\boldsymbol{S_2}$ characterize a low vertex $\Delta K_i'$ of DNS lookups and connections because hosts in stage $\boldsymbol{S_2}$ seek to laterally traverse the local network and escalate privileges whilst establishing a persistent presence in the network. In the same manner, the set of hosts $\{H^9_{int-src}, H^{10}_{int-src}, H^{11}_{int-src}, H^{12}_{int-src}\}$ represent hosts in the infiltration phase $\boldsymbol{S_1}$ with external vertex degrees ΔK_i and clustering coefficient ΔC_i greater than that of internal destinations.

The major differences between the proposed model and other models with respect to modeling and detecting APT attacks are discussed below. This work is compared with five other works that address Advanced Persistent Threats attacks. A total of 6 characteristics, both qualitatively and quantitatively, are put into consideration. Attention is focused on modeling and detecting the different stages of the APT attack process. Additionally, the proposed approach presents an enhanced SNN clustering algorithm that generates clusters from which classes of hosts that make up the top-k ranked hosts are derived. The summarized between the proposed approach and other existing approaches is shown in Table 11.

Apart from the characteristics shown in Table 11, the proposed approach can be used to model and detect the multi-stages APT attack in large-scale network. Dynamic complex network statistics characteristics are used to model and formulate the APT-AN and attack cycle, the open experimental dataset from Los Alamos security lab are used to evaluate the semi-supervised learning approach and model the APT attack life-cycle.

7. Conclusions

To overcome the current limitations of attack network dynamics in APT studies, the use of a semi-supervised learning approach and complex networks properties has been proposed which shows the evolution of the underlying APT-AN. Finite state machines have been used to model the state transitions of the nodes in the time domain. As such, the entire targeted network is modeled as a small-world network and the evolution of the APT-AN as a scale-free network. Therefore, a detection framework based on an enhanced SNN algorithm has been proposed and evaluated. The contributions of this paper are the proposition of a detection framework that scores suspicious Advanced Persistent Threats-related activities at different stages of the APT attack lifecycle and proposition of a semi-supervised learning approach based on an enhanced SNN-based clustering algorithm. Other contributions include modeling of the targeted network as a small-world network model and the evolving APT-AN as a scale-free network model, and application of the proposed framework and semi-supervised learning approach to real network attack dataset of an enterprise network system from the Los Alamos security lab.

Such a model is useful to security and network analysts as a supplement to automated IDS since some aspects of the APT attack cycle cannot be automatically detected. One of the notable challenges in the proposed model is the high computational overhead. Potential next steps and future works include reducing this overhead using cloud computing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 11
Comparison with other works.

Attribute/Model	Applicability to large-scale networks	Dynamic complex network modeling	APT-AN formulation	APT attack cycle formulation	Detection of APT attack stages	Quantitative model evaluation
Ioannou et. al [34]	✗	✗	✓	✓	✗	✓
Oprea et. al [40]	✓	✗	✗	✓	✗	✓
Zhao et. al [15]	✓	✗	✗	✓	✗	✗
Marchetti et. al [21]	✓	✗	✗	✓	✗	✓
Niu et. al [47]	✗	✓	✓	✗	✗	✗
Proposed model	✓	✓	✓	✓	✓	✓

Acknowledgment

This work was supported by the National Social Science Fund of China (18BGJ071).

References

[1] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M.A. Babar, A. Rashid, Data exfiltration: A review of external attack vectors and countermeasures, J. Netw. Comput. Appl. (2018).

[2] A.K. Sood, R.J. Enbody, Targeted cyberattacks: A superset of advanced persistent threats, IEEE Secur. Priv. (2013).

[3] J. Tan, J. Wang, Detecting Advanced Persistent Threats Based on Entropy and Support Vector Machine, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018.

[4] M. Marchetti, A. Guido, F. Pierazzi, M. Colajanni, Countering Advanced Persistent Threats through security intelligence and big data analytics, in: International Conference on Cyber Conflict, CYCON, 2016.

[5] M. Marchetti, F. Pierazzi, M. Colajanni, A. Guido, Analysis of high volumes of network traffic for advanced persistent threat detection, Comput. Netw. 10 (9) (2016) 127–141.

[6] E. Weippl, Advanced persistent threats & social engineering, 2014.

[7] A.M. Juuso, A. Takanen, K. Kittil??, Proactive cyber defense: Understanding and testing for advanced persistent threats (APTs), in: European Conference on Information Warfare and Security, ECCWS, 2013.

[8] A. Lemay, J. Calvet, F. Menet, J.M. Fernandez, Survey of publicly available reports on advanced persistent threat actors, Comput. Secur. (2018).

[9] N. Virvilis, D. Gritzalis, The big four - what we did wrong in advanced persistent threat detection?, in: Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013, IEEE, 2013, pp. 248–254.

[10] S.J. Shackelford, Should your firm invest in cyber risk insurance?, Bus. Horiz. 55 (4) (2012) 349–356.

[11] E. Cole, Advanced Persistent Threat: Understanding the Danger and how to Protect Your Organization, Newnes, 2012.

[12] P. Li, X. Yang, Q. Xiong, J. Wen, Y.Y. Tang, Defending against the Advanced Persistent Threat: An Optimal Control Approach. Hindawi, 2018.

[13] R. Brewer, Advanced persistent threats: Minimising the damage, Netw. Secur. (2014).

[14] N. AbdElatif Mohamed, A. Jantan, O.I. Abiodun, An improved behaviour specification to stop advanced persistent threat on governments and organizations network, in: Lecture Notes in Engineering and Computer Science, 2018.

[15] G. Zhao, K. Xu, L. Xu, B. Wu, Detecting APT malware infections based on malicious DNS and traffic analysis, IEEE Access (2015).

[16] F. Pierazzi, G. Apruzzese, M. Colajanni, A. Guido, M. Marchetti, Scalable architecture for online prioritisation of cyber threats, in: International Conference on Cyber Conflict, CYCON. 2017.

[17] P.K. Sharma, S.Y. Moon, D. Moon, J.H. Park, DFA-AD: a distributed framework architecture for the detection of advanced persistent threats, Cluster Comput. (2017).

[18] A. Oprea, Z. Li, T.F. Yen, S.H. Chin, S. Alrwais, Detection of early-stage enterprise infection by mining large-scale log data, in: Proceedings of the International Conference on Dependable Systems and Networks, IEEE, New York, 2015, pp. 45–56.

[19] A. Alshamrani, S. Myneni, A. Chowdhary, D. Huang, A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities, IEEE Commun. Surv. Tutor. (2019).

[20] S. Singh, P.K. Sharma, S.Y. Moon, D. Moon, J.H. Park, A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions, J. Supercomput. (2019).

[21] M. Marchetti, F. Pierazzi, M. Colajanni, A. Guido, Analysis of high volumes of network traffic for advanced persistent threat detection, Comput. Netw. (2016).

[22] I. Ghafir, et al., Detection of advanced persistent threat using machine-learning correlation analysis, Future Gener. Comput. Syst. (2018).

[23] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, in: The Structure and Dynamics of Networks, 2011.

[24] A.L. Barabási, R. Albert, H. Jeong, Scale-free characteristics of random networks: The topology of the world-wide web, Physica A 281 (1–4) (2000) 69–77.

[25] P. Vixie, What DNS is not, Commun. ACM 52 (12) (2009) 43–47.

[26] T. Kovanen, V. Nuojua, M. Lehto, Cyber threat landscape in energy sector, in: Proceedings of the 13th International Conference on Cyber Warfare and Security, ICCWS 2018, 2018.

[27] B. Nagpal, V. Wadhwa, Cryptoviral extortion: Evolution, scenarios, and analysis, in: Lecture Notes in Electrical Engineering, 2016.

[28] S. Baek, Y. Jung, A. Mohaisen, S. Lee, D. Nyang, SSD-Insider: Internal defense of solid-state drive against ransomware with perfect data recovery, in: Proceedings - International Conference on Distributed Computing Systems, IEEE, 2018, pp. 875–884.

[29] P. Bhatt, E.T. Yano, P. Gustavsson, Towards a framework to detect multi-stage advanced persistent threats attacks, in: Proceedings - IEEE 8th International Symposium on Service Oriented System Engineering, SOSE 2014, IEEE, 2014, pp. 390–395.

[30] X. Fang, L. Zhai, Z. Jia, W. Bai, A game model for predicting the attack path of APT, in: Proceedings - 2014 World Ubiquitous Science Congress: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, DASC 2014, 2014.

[31] R. Yadav, R.N. Verma, A.K. Solanki, Defense-in-depth approach for early detection of high-potential advanced persistent attacks, in: Advances in Intelligent Systems and Computing, 2019.

[32] W. Zhao, P. Wang, F. Zhang, Extended petri net-based advanced persistent threat analysis model, in: Lecture Notes in Electrical Engineering, 2014.

[33] A. Zimba, H. Chen, Z. Wang, Bayesian network based weighted APT attack paths modeling in cloud computing, Future Gener. Comput. Syst. (2019).

[34] G. Ioannou, P. Louvieris, N. Clewley, G. Powell, A Markov multi-phase transferable belief model: An application for predicting data exfiltration APTs, in: Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, 2013.

[35] A.M. Lajevardi, M. Amini, A semantic-based correlation approach for detecting hybrid and low-level APTs, Future Gener. Comput. Syst. (2019).

[36] D. Yan, F. Liu, K. Jia, Modeling an information-based advanced persistent threat attack on the internal network, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–7.

[37] B. Binde, R. McRee, Assessing outbound traffic to uncover advanced persistent threat, retrieved from sans inst. web, 2011.

[38] F. Labs, Fireeye Advanced Threat Report 2013, FireEye Labs, 2013, [Online]. Available: https://www2.fireeye.com/rs/fireye/images/fireeye-advanced-threat-report-2013.pdf.

[39] E. Hutchins, M. Cloppert, R. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, in: 6th International Conference on Information Warfare and Security, ICIW 2011, 2011.

[40] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, D. Ghosal, SIDD: A framework for detecting sensitive data exfiltration by an insider attack, in: Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS, 2009.

[41] S. Wen, Y. Rao, H. Yan, Information protecting against APT based on the study of cyber kill chain with weighted bayesian classification with correction factor, in: ACM International Conference Proceeding Series, 2018.

[42] J. Grier, Detecting data theft using stochastic forensics, in: DFRWS 2011 Annual Conference, 2011.

[43] T.F. Yen, et al., Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks, in: ACM International Conference Proceeding Series, 2013.

[44] T. Sasaki, Towards detecting suspicious insiders by triggering digital data sealing, in: Proceedings - 3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS 2011, 2011.

[45] P. Chen, L. Desmet, C. Huygens, A Study on Advanced Persistent Threats, Springer, Berlin, Heidelberg, 2014, pp. 63–72.

[46] C. Velazquez, Detecting and preventing attacks earlier in the kill chain, 2015.

[47] W. Niu, X. Zhang, G. Yang, R. Chen, D. Wang, Modeling attack process of advanced persistent threat using network evolution, in: IEICE Transactions on Information and Systems, 2017.

[48] B.S. Dohleman, Exploratory social network analysis with pajek, Psychometrika (2006).

[49] A.L. Barabási, R. Albert, H. Jeong, Scale-free characteristics of random networks: The topology of the world-wide web, Physica A (2000).

[50] A. Zimba, Z. Wang, H. Chen, Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems, 4 (1) (2018).

[51] A. Kent, Comprehensive, Multi-Source Cyber-Security Events, Cyber Security Science, Los Alamos National Lab. (LANL), Los Alamos, NM, 2015.

[52] B. Nagpal, V. Wadhwa, Cryptoviral extortion: Evolution, scenarios, and analysis, Lect. Notes Electr. Eng. (2016) 309–316.

[53] M.L. Goldstein, S.A. Morris, G.G. Yen, Problems with fitting to the power-law distribution, Eur. Phys. J. B 41 (2) (2004) 255–258.

[54] A. Oprea, Z. Li, T.F. Yen, S.H. Chin, S. Alrwais, Detection of early-stage enterprise infection by mining large-scale log data, in: Proceedings of the International Conference on Dependable Systems and Networks, 2015.

[55] L. Ertoz, M. Steinbach, V. Kumar, A new shared nearest neighbor clustering algorithm and its applications, …Data Appl. … 2 (1) (2002) 105–115.

[56] L. Ertöz, M. Steinbach, V. Kumar, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, 2003.

[57] H.B. Bhavsar, A.G. Jivani, The shared nearest neighbor algorithm with enclosures (SNNAE), in: 2009 WRI World Congress on Computer Science and Information Engineering, CSIE 2009, 2009.

[58] C.M. Bishop, Pattern Recognition and Machine LeBishop, C. M. (2006) Pattern Recognition and Machine Learning, Pattern Recognition, Springer Publishing, Cambridge CB3 0FB, 2006, http://dx.doi.org/10.1117/1.2819119.arning.

[59] J. Cao, H. Wang, D. Jin, J. Dang, Combination of links and node contents for community discovery using a graph regularization approach, Future Gener. Comput. Syst. (2019).

[60] C. Canali, M. Colajanni, R. Lancellotti, Hot set identification for social network applications, in: Proceedings - International Computer Software and Applications Conference, 2009.

[61] W. Jiang, G. Wang, J. Wu, Generating trusted graphs for trust evaluation in online social networks, Future Gener. Comput. Syst. 31 (1) (2014) 48–58.

[62] L. Al Shalabi, Z. Shaaban, Normalization as a preprocessing engine for data mining and the approach of preference matrix, in: Proceedings of International Conference on Dependability of Computer Systems, DepCoS-RELCOMEX 2006, 2007.

[63] J.P.S. Kevin A. Wilson, Nathan D. Green, Laxmikant. Agrawal, XibinGao, Dinesh. Madhusoodanan, Brian. Riley, Laxmikant agrawal xibingao dinesh madhusoodanan brian riley graph-based proximity measures, Pract. Graph Min. R (2013) 135–165.

[64] E. Maeland, On the comparison of interpolation methods, IEEE Trans. Med. Imaging (1988).

[65] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. (2006).

Aaron Zimba is lecturer at Mulungushi University and obtained his PhD in Network and Information Security at the University of Science and Technology Beijing in the Department of Computer Science and Technology. He received his Master and Bachelor of Science degrees from the St. Petersburg Electrotechnical University in St. Petersburg in 2009 and 2007 respectively. He is also a member of the IEEE. His main research interests include Network and Information Security, Network Security Models, Cloud Computing Security and Malware Analysis.

Hongsong Chen received his PhD degree in Department of Computer Science from Harbin Institute of Technology, China, in 2006. He was a visiting scholar in Purdue University from 2013–2014. He is currently a professor in Department of Computer Science, University of Science and Technology Beijing, China. His current research interests include wireless network security, attack and detection models, and cloud computing security.

Zhaoshun Wang is a Professor and the Associate Head of the Department of Computer Science and Technology at the University of Science and Technology Beijing. He graduated from Department of Mathematics at Beijing Normal University in 1993. He received his PhD from Beijing University of Science and Technology in 2002. He completed postdoctoral research work at the Graduate School of the Chinese Academy of Sciences in 2006. He holds patents and has many awards to his name. His main research areas include Information Security, Computer Architecture and Software Engineering.

Mumbi Chishimba holds a Master's and Bachelor's Degree in Computer Science from Mulungushi University in the department of Computer Science and Information Technology. Currently, he is with the National Institute of Public Administration (NIPA) where is he is serving as the information systems analyst and developer. His research interests are information systems management, software algorithm development, and information and network security.