International Journal of Security and Its Applications
Vol. 12, No. 4 (2018), pp.15-28
http://dx.doi.org/10.14257/ijsia.2018.12.4.02

A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention

Ayei E. Ibor[1]*, Florence A. Oladeji[2] and Olusoji B. Okunoye[3]

[1]*Department of Computer Science, University of Calabar, Calabar, Nigeria*
[2,3]*Department of Computer Sciences, University of Lagos, Lagos, Nigeria*
ayei.ibor@gmail.com, foladeji@unilag.edu.ng, bokunoye@unilag.edu.ng

Abstract

The expanding threat landscape has come with a plethora of consequences for most organizations and individuals. This is witnessed in the high volume of cyber-attacks prevalent in the cyberspace. Though several approaches have been proposed and deployed in recent times, most of them are only theoretical masterpieces while others remain computationally infeasible due to the computational requirements for implementing them. Where implementation is possible, the issue of computational complexity becomes a significant overhead in which case a large amount of computing resources such as CPU cycle, memory, network bandwidth and data structures are consumed culminating in tedious, time consuming, and error prone processes. Similarly, most of these techniques are basically reactive and as such can only be triggered when an incident is reported while most depend on the administrator to apply the necessary mitigation processes after an attack has occurred. To this effect, this paper presents a survey of cyber security approaches that have been proposed in the Literature. The paper also reviews the methodologies, strengths and weaknesses of these approaches, and finally identifies areas where further research could focus.

Keywords: *cyber security, cyber-attacks, machine learning, network intrusions, information security*

1. Introduction

Cyber security is a major concern for a large number of organisations, institutions, corporations and individuals across the globe. Buczak and Guven in [1] assert that the totality of the technologies and processes for the monitoring and prevention of unauthorized access, alteration, misuse, and denial of service to computer networks and resources constitute cyber security. This also includes the tendency to authorize access to classified contents, and critical infrastructure, which are network-accessible.

Most networks are widely connected to each other through the Internet, and provide a means for sharing data, information, intelligence, software, and hardware. Though, the sharing of valuable resources for enhanced operational efficiency has characterised the computer networking paradigm, it has also created a seamless source of easy propagation of malware, and through this, the escalation of cyber-attacks has become prevalent in the cyberspace.

This expansion in the threat landscape is a factor of the growing power of cyber force, which is gradually creeping into the control of all domestic, business and industrial functions. As a consequence of the effect of cyber force, Akyazi in [2] asserts that the dangers of cyber-attacks come with the ability to modify the parameters of a system or database in order to generate a kinetic effect for escalating attacks including the tendency to destroy classified contents.

Received (February 21, 2018), Review Result (May 16, 2018), Accepted (May 18, 2018)

ISSN: 1738-9976 IJSIA

Protecting against cyber-attacks requires both proactive and reactive approaches. These approaches, which can also be described as active and passive are relevant in the context of use – basically direct defensive actions or mitigation techniques against cyber threats. Denning in [3] mentioned that the relevance of cyber defence strategies is embedded in the capacity to truncate active and passive threats, which have become a norm in the cyber domain.

It is therefore pertinent to understand the research gaps in the current cyber security approaches. To this effect, this paper will highlight the numerous techniques available in the public domain including the strengths and weaknesses of each. The succeeding sections will discuss cyber security approaches in terms of detection, prediction, and prevention.

Generally, attack detection and prediction can be achieved using machine learning and evolutionary algorithms, as well as statistical techniques and association rules. Similarly, most attack prevention approaches are achieved through traffic analysis to identify and drop (or block) a malicious activity. This is depicted in Figure 1.

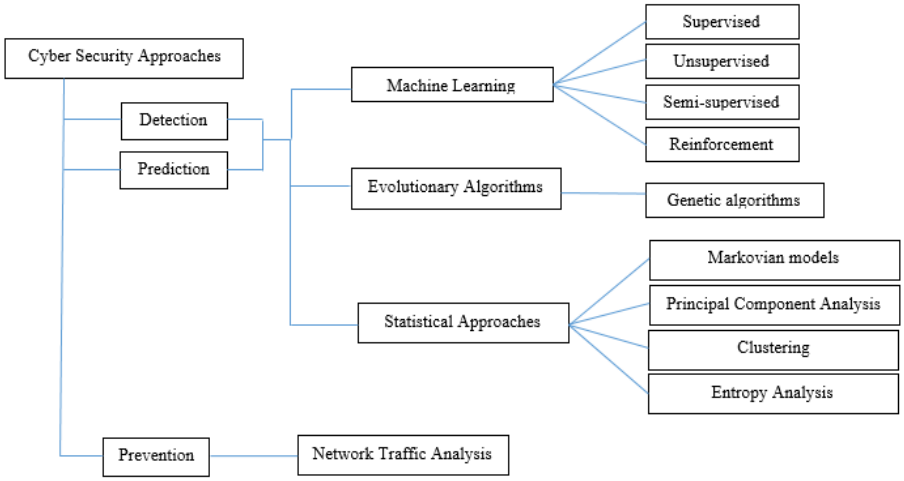

Figure 1. Summary of Cyber Security Approaches

2. Cyber-attack Detection Approaches

Cyber-attacks detection is a common attack mitigation technique. It involves responding to an abnormal connection to report the presence of an attack pattern or profile in a network. One of the major approaches to detecting cyber-attacks is intrusion detection. According to [4], intrusion detection is the process of identifying an intrusion or attack signature in a continuous flow of connections. Intrusion detection is achieved with the use of intrusion detection systems.

Intrusion detection systems are categorized into three approaches. These include misuse (signature-based), anomaly, and hybrid detection approaches respectively. While misuse detection considers the signatures of known attacks to help in detecting intrusions, anomaly detection uses profiles of normal network activities to flag intrusions where a deviation from the normal profile is detected. The combination of the two approaches results in a hybrid approach [5].

There are several literatures on cyber-attack detection in the public domain. However, most of these approaches have been largely inefficient in detecting attacks while some have resulted in the high consumption of computing resources. Similarly, most of the approaches proposed in the extant literature are computationally infeasible, and can only remain as theoretical masterpieces. Subsequent sections will discuss more cyber-attack

detection approaches in the public domain, and will also highlight the methodology, strengths and weaknesses of each approach.

2.1. Detection by Machine Learning Approach

Machine learning techniques have become popular in detecting cyber-attacks in recent times. Machine learning is particularly efficient for analyzing data and predicting the outcome of certain events based on the available sample inputs, which are used to build a suitable model for making the right decisions [1]. The main tasks of machine learning algorithms are to classify and predict the presence or absence of a learned instance using training data. The application of machine learning in the current scenario of cyber-attack detection has helped to improve the detection process to a high degree of accuracy.

Four types of machine learning techniques are discussed in this paper. These include supervised, unsupervised, semi-supervised and reinforcement learning techniques.

i. Supervised Learning Approach

Supervised learning is an aspect of pattern recognition, which uses a set of labelled instances considered as training data with the corresponding desired output. With the labelled instances, a predictive model is derived at the training phase to classify new datasets. This is achieved by feeding the labeled instances into a certain machine learning algorithm. Several of these machine learning approaches as mentioned in [1] include Decision Trees such as C4.5 and ID3 algorithms, Artificial Neural Network, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Hidden Markov Model (HMM), and Naïve Bayes.

One aspect of the cyberspace that is prone to malicious attacks is web pages. With an ever expanding web presence in all categories of devices, and the constant use of the contents of web pages for such activities as social media interactions, e-commerce, online banking, e-government, and several others, the need for an efficient approach for detecting malicious web pages cannot be overemphasized.

The possibility of easily modifying the source code of web pages by injecting malicious code as witnessed today can result in a new class of malicious web pages, which can escalate the attack landscape and deceive users into divulging critical personal information. To this effect, [6] proposed a hybrid technique for detecting malicious web pages using misuse and anomaly detection approaches. In their paper, misuse detection was achieved with C4.5 decision tree algorithm while anomaly detection, which allowed for the detection of new instances of malicious web pages used a one - class support vector machine (SVM).

The hybrid malicious web page detection technique hierarchically combines the misuse and anomaly detection modules such that at the first instance, every web page undergoes analysis by the misuse detection module. This module then uses the decision tree algorithm to detect malicious web pages by matching the properties of these pages with known patterns of web pages. At the completion of the first stage, the unclassified pages are fed into the anomaly detection module in order to detect new instances of malicious pages with the help of the one-class SVM.

The approach produced an improved detection rate of 98.8% by combining the misuse and anomaly detection approaches, which complement each other in the process of detecting known and unknown instances of malicious web pages. While the approach was able to achieve a detection rate of 98.8%, it produced a false positive rate of 30.5%, which is a major drawback of the proposed approach. It is assumed that the high false positive rate is a function of the increased similarity rate of input data to the anomaly detection component resulting in the flagging of rather benign web pages as malicious.

In [7], a combination of supervised learning algorithms such as C5.0 decision tree and one class SVM algorithms are used to achieve a hybrid technique for the classification of

anomalous and normal activities in a network. The approach is underpinned by the building of the misuse detection model with C5.0 decision tree and from the information acquired, the anomaly detection model is achieved using the one class SVM. The proposed technique also relied on the cuttlefish algorithm for feature extraction to reduce redundancy and enhance the detection accuracy as well as the false alarm rate.

The cuttlefish algorithm (CFA) finds the best optimal subset from the sample dataset (NSL-KDD dataset) at the end of the data preparation and preprocessing phase. This sample dataset is partitioned into the training and test data. The best selected subset is fed into the C5.0 algorithm to generate the node information in connection with the original data. This node information subsequently becomes useful for the one class SVM to perform the final classification of the attack data as anomalous or normal.

With the hybrid approach underpinned by the use of both the misuse and anomaly detection techniques for detecting network attacks, performance was enhanced including a reduction in the time complexity, also producing a higher detection accuracy of up to 98.2% and low false alarm rate of 1.7%. Using a decision tree algorithm for predicting instances in this case comes with its own limitations. Decision trees can be unstable when the precise information is not used, and as such a minor change in the input data can produce significant changes in the tree. This is not ideal for detecting intrusions as the resultant classification can be completely erroneous with dire consequences to critical network infrastructure.

A 3-step process for the realization of a novel feature representation technique based on the cluster centre and nearest neighbour (CANN) approach is given in [8]. This approach uses two measured and summed distances, which include in the first instance, the distance between the data point and its cluster centre, while the second distance is considered in terms of the data point and other nearest neighbours within the same cluster.

The resultant one-dimensional distance based feature is used to depict each data point in the chosen sample space to achieve attack detection using the k-Nearest Neighbour (k-NN) classifier. In the first step, a clustering algorithm is used to extract cluster centres, and the cluster count is a function of the number of classes from the given dataset.

At the second stage, a one new dimension feature is generated to depict a data point by measuring and summing the distances in two perspectives viz-a-viz between the data points in the dataset and the cluster centres, and as well an individual data point and its nearest neighbours in a similar cluster. Finally, there is the extraction of cluster centres and nearest neighbours to formulate new data. With the combination of the test and original training sets, the k-NN classifier is trained and tested to detect new and unknown instances in the connection strings.

In the experimentation, it was observed that the CANN approach was efficient as compared to k-NN and SVM classifiers with respect to the used six dimension dataset, and demonstrated a significantly high detection accuracy with a low false positive rate. Conversely, CANN achieved the same rate of performance as k-NN and SVM classifiers when considering the nineteen dimension dataset.

Some of the limitations identified in the approach include the inability of CANN to detect user to root (u2r) and root to local (r2l) attacks. This may not be unconnected with the use of a one dimensional distance based feature representation to train and test the model that eventually detects the different classes of attacks. In doing so, it is assumed that the feature space cannot exhaustively represent the patterns of u2r and r2l attacks.

A multiple learning technique that considers an improvement to the cluster center and nearest neighbour (CANN) approach as discussed in [8] is further clarified in [9]. The approach, called ICANN, deploys two supervised machine learning algorithms, that is, k-Means classification algorithm and k-Nearest Neighbour algorithm.

The stages of the approach include data preprocessing, k-Means clustering, training and classification. Normalization or preprocessing linearly transforms the data based on minimum and maximum feature set. This process is immediately followed by the

clustering of the preprocessed dataset using the k-Means clustering algorithm. The data is clustered into test and training datasets by the assignment of the most similar data to a certain cluster. Five (5) clusters are generated based on the number of attack types (4) and one (1) normal connection in the NSL-KDD dataset.

To further improve the model, using the five types of behavioural profiles defined, that is, one normal and four attack profiles, the k-Nearest Neighbour algorithm is used to locate the nearest neighbour of each data point to its cluster. The distance between a data point and its nearest neighbour is determined based on Euclidean distance with the maximum distances chosen. The next distances calculated are between the data points and the five cluster centres using Euclidean distance formula. The calculated distances are considered sufficiently large enough to distinguish between normal and abnormal data, and the generated similarities and differences between the test and training data are then applied to detect attack patterns in new instances of data.

As reported in the paper, ICANN produced a higher accuracy as well as detection rate. A lower false alarm rate was also reported, and the ability to detect u2r and r2l attacks as compared to CANN, though ICANN required more computational effort than CANN.

ii. Unsupervised Learning Approach

Unsupervised learning works by discovering patterns in an unlabeled dataset used as the training data in order to make the rightful classification decisions in a set of new instances. This usually involves the use of clusters to identify the classes to which instances belong. Song *et al.*, [10] discussed an anomaly detection system with an unsupervised learning approach that is able to tune and optimize the value of parameters automatically in order to arrive at better classified instances, which either represent an attack string or a normal connection.

The proposed approach performs the classification of instances after the training phase, which comprises such stages as filtering, clustering, and modeling. Filtering achieves the required subset of normal data, which is then partitioned into k clusters. These k clusters represent normal patterns in the traffic data such as HTTP, SMTP, and FTP. For each normal cluster, the one class SVM is used to generate k SVM models also called k hyperspheres for classification. Each k model is then matched with new instances to determine whether such an instance occurs within the predefined hypersphere, in which case it is a normal connection, otherwise an attack state is flagged.

The use of unsupervised learning in the approach provides an effective technique for classifying new instances by using a threshold to define attack and normal data at the time of building the model. At this point, a significant drawback of the approach can be vividly identified based on the fact that normal connections vary on heterogeneous networks, and as such building profiles of normal behaviour can deteriorate significantly. This significant deviation in the behavourial patterns and characteristics of one network with respect to other networks can lead to an inefficient model, which will invariably require a good measure of parameter tuning and optimization to suit the requirements of a specific network environment.

Abduvaliyev *et al.*, [11] and Butun *et al.*, [12] investigated the various attacks on wireless sensor networks (WSNs), and the effect approaches in attack detection can have on the expanding threat landscape. However, [11] posits that such attacks as denial of service (DoS), sinkhole/blackhole attacks, selective forwarding, node replication attacks, and wormhole attacks require a second line of defence for protecting against their impact on a WSN. This second line of defence is based on a clustering technique considered as an unsupervised learning algorithm, which helps to detect anomalous traffic in WSNs. The model is built based on twelve network traffic patterns, which are then employed in the training and testing stages.

At the point of training the model, a fixed-width clustering algorithm creates clusters in the feature space. Anomalous clusters are identified if such samples contain less training

traffic samples than a specific threshold. Furthermore, matching a certain traffic sample to a cluster set is performed at the testing stage to report the presence of an anomalous pattern or not. One significant drawback of this method is the high computation requirements on sensor nodes, which can result in significant overheads on the host network.

iii. Semi-supervised Learning Approach

Ashfaq *et al.*, [13] posit that semi-supervised learning considers both labeled and unlabeled samples to achieve a better classifier. Similarly, [4] claims that semi-supervised machine learning approach models normal behaviour with the help of a pre-labelled dataset. Consequently, semi-supervised learning combines the power of both supervised and unsupervised learning approaches in the process of building a model for classifying new instances of a dataset.

Furthermore, [4] proposed a two-stage semi-supervised statistical approach for detecting network anomalies. The technique builds a probabilistic model using pre-labeled normal instances. This model is then used to evaluate deviation from the normal behaviour using a predetermined threshold. The second stage uses an iterative process to reduce the false alarm rate leveraging a similarity distance and dispersion rate of the initial classifications of the probabilistic model [4].

With this approach producing good results in terms of a high detection rate and low false positive rate, and clearly outperforming the Naïve Bayes algorithm in terms of true positive and false positive rates, it is still bound by the limitations of the anomaly detection approach as discussed in [5].

Han *et al.*, [14] proposed a semi-supervised learning approach as a countermeasure for co-resident attacks in cloud based environments. The approach achieved a defensive mechanism that makes it computationally expensive for a co-resident attack to succeed in a virtual machine in a cloud computing system. The problem was modeled as a 2-player security game with users classified using clustering analysis and semi-supervised SVMs. The users are considered as high risk (malicious), medium risk (uncertain), and low risk (legal) in tandem with the modification of the virtual machine allocation process. This helps the defensive mechanism to escalate an attacker's overall cost to achieve a computationally expensive attack process.

The approach recorded success by escalating the overall cost of the attacker up to two orders of magnitude as a countermeasure for co-residence attacks. This notwithstanding, the use of a single datacenter to implement the approach is not feasible in practice for describing the different scenarios in multiple datacenters, which are likely to support co-location and co-resident attacks.

In [13], a novel fuzziness based approach that uses semi-supervised learning is proposed. The approach considers unlabeled samples with the assistance of a supervised learning algorithm to enhance the performance of the classifier for intrusion detection. Furthermore, a fuzzy membership vector becomes the output of a trained single hidden layer feed-forward neural network (SLFN). With this membership vector, samples are classified as low, mid, and high fuzziness categories on the unlabeled samples with the fuzzy quantity.

The model built is further retrained using each fuzziness category in relation to the original training dataset to achieve a better classification of new instances with an average accuracy of 83.27% for the first dataset containing 22, 544 records, and 67.94% for the second dataset with 11, 850 records. The proposed approach performed better than such classifiers as J48, Naïve Bayes, Random Forests, Random Tree, and SVM. On the other hand, the approach is constrained by dependence on determining the relationship between the output of the classifier in terms of its fuzziness on a set of samples and their misclassification rate.

Additionally, the classification was based on a two-class problem represented as normal and anomaly classes, and did not consider the detection of multiple types of attacks.

iv. Reinforcement Learning Approach

Reinforcement learning is a machine learning approach that allows a software agent such as a sensor node to learn by interacting with its environment. Alsheikh *et al*., [15] posits that reinforcement learning is important in the context of pattern recognition as it allows software agents to create experiences from their interactions with the environment in order to take the best actions for long-term rewards. Similarly, [31] mentioned that reinforcement learning agents pass messages within an initially unknown environment and uses the information gained to redefine action policies to maximize their rewards.

In [16], reinforcement algorithms are discussed. The authors posit that reinforcement learning is suitable for solving sequential problems, which can be modeled as Markov decision processes (MDPs), and as such suitable for interpreting learning control problems. These problems are usually difficult to interpret by supervised learning algorithms. A typical reinforcement learning problem is depicted in Figure 2.

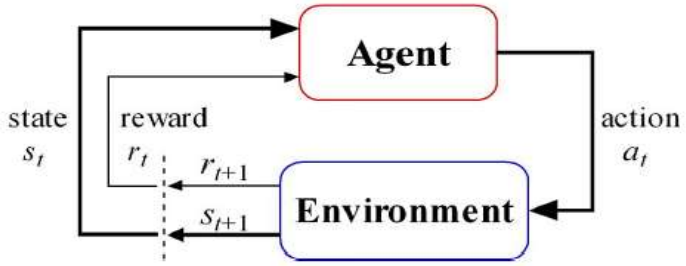

Figure 2. A Model of Reinforcement Learning Problem [16]

Shamshirband *et al.*, [17] applied fuzzy Q-learning for detecting and preventing intrusions in WSNs. The technique uses a combination of cooperative game theory and fuzzy Q-learning algorithms in order to detect DDoS attacks. The approach models sink holes, a base station and an attacker in a 3-player strategy game such that the game is activated when a flood of packets is directed at a victim node. At this point, the packets received are measured against a specific alarm event threshold in WSN and when such a threshold is breached, the approach implements cooperative defense countermeasures for the sink hole and the base station.

For the evaluation of performance, low energy adaptive clustering hierarchy (LEACH) was simulated with NS-2 simulator to demonstrate the detection and defense accuracy of the approach. The architecture of the approach allows the sink hole and base station to adapt in the process of selecting the most appropriate strategy to detect and respond to a spontaneous attack. To detect future attacks, the IDPS regularly modifies its learning parameters using fuzzy Q-learning in a process described as continuous self-learning of past attacks. With the approach only considering DDoS flooding attacks, it may be difficult to ascertain its effectiveness against other forms of attacks. Consequently, the model requires a holistic improvement to effect enhanced decision making capabilities especially with respect to truncating novel attacks.

2.2. Detection by Genetic Algorithms Approach

Genetic algorithms (GAs) are a composite part of evolutionary algorithms (EAs), which are basically metaheuristics defined by the process of natural selection. The most important function of a genetic algorithms is anchored in generating solutions to

optimization and search problems based on such bio-inspired operators as mutation, crossover, and selection. Consequently, [18] proposed an intrusion detection system that relies mainly on the use of a genetic algorithm.

The genetic algorithm approach is tuned to detect various forms of attacks based on evolution theory to information evolution with a consequence to filter the captured traffic data in order to minimize the complexity attributable to classification. The effectiveness of the approach was measured using three factors, which include the fitness function, the representation of individuals, and the parameters of the GA.

To achieve the purpose of the algorithm, two main stages are used in the proposed approach. These include the pre-calculation and detection phases. In the pre-calculation phase, the training data is used to create a set of chromosomes, which is used for comparisons in the next stage. In the second stage, detection is achieved by creating a population for testing the approach and the test data is eventually predicted using some evaluation processes such as selection, crossover, and mutation. A fitness function then computes the fitness of each chromosome of the sample population.

Results from experimentation showed that the approach performed well against various types of intrusions including probe, user to root (U2R), root to local (R2L), and denial of service (DoS) attacks. Measuring the fitness of a chromosome with the standard deviation equation with distance limited the performance of the approach in terms of the detection and false positive rates. Using a more efficient heuristic in this regard can be very valuable for an improved detection system.

This conceptualization of the structure and processing of a genetic algorithm as adopted from [18] is illustrated in Figure 3.

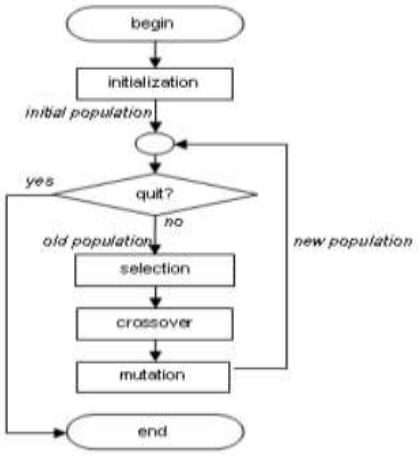

Figure 3. Processes of a Genetic Algorithm

Ahmad *et al.*, [19] proposed a detection technique based on genetic principal components towards improving the performance of support vector machine (SVM). The approach applied the optimal subset feature selection and discriminatory power to reduce redundancy in data in order to enhance the classification of attack instances by SVM

The proposed approach selects a dataset as input. This dataset undergoes a process called feature transformation, which generates more visible and discriminant features in the principal space. Principal component analysis (PCA) is applied to the feature space to transform it such that redundancy is eliminated in the resultant feature subset for classification. The actual selection of the feature subset is performed by a genetic algorithm, which searches the PCA space to choose a subset of principal components otherwise referred to as genetic principal components.

The classification is then achieved with the SVM classifier, and this process is followed by the training and testing of the model. During training, the model is tuned to

find optimal parameters while testing is used to evaluate the trained model to classify network connections as normal or intrusive. Results obtained from the experimentation showed the ability of the proposed approach to minimize the number of features and at the same maximize the detection rate, producing an average detection rate of 99% as compared to other approaches.

Although the performance of SVM is enhanced in this approach, SVM being a supervised learning algorithm requires that all input data be fully labelled in all cases. This is likely to introduce latency at the training phase of the model, and can result in the difficulty to interpret parameters of the solved model.

Rastegari *et al.*, [20] considered evolving statistical rulesets for attack detection. The approach is underpinned by the need to carefully analyse network traffic data owing to the similarity between normal and attack patterns in order to explicitly detect intrusive traffic using a genetic algorithm. The algorithm is tuned to evolve simple interval-based rules using statistical continuous-valued input data. Each rule is then evaluated using a fitness function in conjunction with a new representation of individuals. In this way, improved rulesets are generated during the learning phase. The generated rules are subsequently used for the classification of the data points.

In the model, the chromosome structure is configured to accommodate rules with mutable feature set and a fitness function that is able to reward cooperation between rules. This also includes a mechanism for adaptively assessing the degree of exclusiveness at the selection stage in order to generate succinct rulesets.

The normalization of the dataset is performed during the preprocessing stage to produce normal and attack records for training and testing the evolving rule-based classifier. Preprocessing also considers an optional feature selection stage that works to provide seed rules for the initial rule population. This is followed by the evaluation phase, at which point a traditional fitness function assesses component rules while a higher level function selects rulesets that work together in the detection process.

One major strong point of the proposed approach is the non-dependence on categorical features of packet headers such as source and destination IP addresses. This implies that the approach leverages statistical features of network traffic for detecting any available suspicious behaviour in the traffic pattern, and as such suitable for detecting novel attacks. Similarly, the use of succinct rulesets, which are evaluated by the genetic algorithm, and found to cooperate while covering the area of search precisely, keeps the rulesets small and efficient in detecting known and novel attacks.

Considering the number of rulesets used in classifying normal and attack instances as well as the fitness and performance measurements, it is pertinent to consider miniature modifications in the boundary values. Unfortunately, the proposed model is insensitive to these modifications since no training samples are proximate making it difficult to be deployed in a multi-class situation for identifying the type of attack infiltrating a network at any point in time.

3. Cyber-attack Prediction Approaches

Cyber-attack prediction involves the projection of the likelihood of an attack on a controlled and dynamic network environment. According to [21], the main goal of cyber-attack (intrusion) prediction is to enhance the security capabilities of defence systems in the cyberspace. To this effect, GhasemiGol and Ghaemi-Bafghi in [22] proposed an entropy-based alert correlation system called E-correlator to simplify the analysis of a large set of alerts in such a way that there is no information loss between the correlated and the original raw alerts. The E-correlator system takes raw alerts as input and generates a hyper-alerts graph as output. This process is achieved with the use of the density-based spatial clustering of applications with noise (DBSCAN) algorithm.

The use of DBSCAN algorithm in aggregating similar alerts to specific clusters comes with a couple of benefits such as not specifying the number of clusters in the data a priori, efficient in locating randomly shaped clusters and as well identifying noise or outliers in the data set. The proposed model also has a log linear computational complexity of O(n log n), which is reasonably fast enough for such a large data set.

However, the E-correlator system did not consider other sources of intrusions such as vulnerabilities in applications, services and protocols of host, and as such its efficacy is likely limited in scope, and cannot be used to generalize the process of alert correlation and clustering for multistage attack prediction. In another perspective, the DBSCAN algorithm, though resistant to noise and outliers, is prone to poor cluster descriptors as well as high sensitivity to input parameter setting.

The approach of [23] models the prediction of attacks using critical episodes that overrun an episode window, and are consequently used to construct an attack tree. The construction of an attack tree in modeling attack scenarios in their approach including the detection and prediction of multistage attacks produced a 95% accuracy in both cases.

Constructing attack trees can be simple to learn and use, and as well can produce a clear output. Attack trees can be used to model the decision process of the attacker. This is done by constructing a tree that has as a root node the attacker's goal while the leaf nodes depict the different paths through which such a goal can be accomplished [23].

The use of an attack tree for the reconstruction of attack strategies gives a vivid structured view of the events prior to the attack. Attack trees are useful for tracing undesired events to their likely causes in order to determine the goal of an attack or a certain security incident. The time consuming nature of attack trees construction is a major concern in this approach. At the same time, the use of attack trees makes the approach less scalable for large networks since only the modeled attack plans in the library can be used.

A hybrid approach for anomaly intrusion detection with a prediction module is studied in [24]. One of the key components of multistage attacks is the preponderance of network anomalies, which can populate the network server to the extent of exhausting the available computing resources such as memory, processors, data structures, and storage.

The proposed approach works on a combination of data mining techniques such as the k-means clustering algorithm and radial basis kernel function (RBF) of the support vector machine (SVM). The RBF kernel function is used for classification, which focused on decreasing the attribute count associated with each data point to improve the rate of detection and accuracy when applied to the KDDCup'99 dataset.

The approach involves feature selection to create a subset of training data from the original feature space. The essence of feature selection in this respect is to reduce the computational complexity of the use of the entire feature space in the dataset. Different features were selected for each class of attack to allow for accurate prediction of instances in the modeled attack space. The results showed that the hybrid approach returned better results leveraging all 41 features of the KDDCup'99 dataset than each of the k-means clustering algorithm and support vector machine (SVM) with an accuracy of 80.28.89% for all classes of attacks as compared to 73.24% for k-means and 49.3% for SVM.

In addition, the accuracy for DoS attacks was high, ranking up to 93.33% as compared to the 86.67% and 40% for k-means and SVM approaches. For a selected feature space, the hybrid technique reported an accuracy of 100% for DoS and 87.01% for all classes of attacks with a significantly lower false alarm rate. The results represents a good performance for the hybrid approach proposed in [24].

One significant drawback of the approach is that the k-means clustering algorithm used is to a measureable extent, sensitive to noise and outliers. This is because a small quantity of the data points can considerably have effect on the centroids. Most importantly, [25] highlighted that RBF results in a low order of accuracy in practical use. This is likely to affect the efficiency of the proposed hybrid technique in [24].

4. Cyber-attack Prevention Approaches

Prevention of attacks is a proactive activity that identifies and responds to potential threats in a network quickly. Prevention is extremely relevant in the process of mitigating cyber-attacks. Most detection approaches are reactive, and are only applied after much damage has been done on the impact zone. Several intrusion prevention systems (IPSs) have been proposed as a means of improving the security of the cyberspace.

Patil and Meshram [26] discussed an approach for preventing network intrusions with emphasis on variants of DoS attacks such as flooding, flooding using IP spoofing, and ping of death attacks. The technique is modeled to be platform independent using the java virtual machine (JVM). The packet sniffer is developed using Jpcap library while the prevention of malicious traffic from infiltrating the internal network is achieved with the use of Linux iptables command. The use of the Jpcap library is preceded by the installation of the winpcap (for windows) and libpcap (for Unix or Linux) libraries.

Attack prevention, in this case, is performed by examining inbound packets captured using Jpcap in promiscuous mode. When the packets are examined and the SYN flag in the packet is set and pointing to the same destination address over a continuous flow of network traffic, the system assumes a SYN flood attack. The detected attack information is stored in the log file, and further actions are taken to drop the packet with the execution of the iptables command (Linux) or net-filter (Windows).

Other attacks that can be prevented by the proposed system include smurf attack (ICMP packets), SYN-FIN attack, XMAS attack, fraggle attack (UDP packets), and all flag attacks. The approach delivered a quick process for attack prevention, though largely dependent on iptables rules. Moreover, when such an attack is not detected early enough through a quick rule-match process, such an attack can still infiltrate the network, and cause a lot of damage to the internal network resources.

Several types of attacks and their variants have been mentioned in this work including the approaches used for mitigating them. However, one of the most evasive attack types in the cyberspace is Distributed Denial of Service (DDoS) flooding attack, which uses Botnets (otherwise called an attack army) to disrupt services provided for genuine users of a system or network. Botnets are commonly deployed to flood a large number of computers, sometimes, on a global scale with malicious packets over the Internet by exploiting security holes also called vulnerabilities in these computers [27].

These botnets are made of components called masters, handlers, and bots (which are the agents that distribute the malformed packets for a successful DDoS attack). The attackers constitute the masters, who communicate with the agents or bots using handlers. The attackers provide the command and control using zombies. Zombies are already compromised computers, which are part of the attack army, randomly compromising other vulnerable computers in the attack path in order to escalate the attack state until all computing resources are completely debilitated at which point severe and irreversible damages can be done.

Defending against DDoS attacks has been a herculean task. Ideally, hundreds and thousands of computers connected to Internet-enabled networks as well as other devices such as smart phones and personal digital assistants (PDAs) should be protected from all forms of vulnerable services and ports. However, these vulnerable services and ports arise from unpatched systems, for which most security updates pushed to devices through proxy servers are not applied, and as when due. This creates a myriad of unpatched and vulnerable machines, which can be exploited during a DDoS flooding attack using the command and control (C & C) capability of botnets.

Consequently, [27] posits that the use of such mechanisms as source address authentication, capability and filtering mechanisms are relevant in the context of addressing DDoS flooding attacks. Similarly, Internet service providers should collaborate, using such technologies as cloud computing and IoT, in the process of

preventing and truncating such attacks using the proximity of the Internet as a major boost in this regard.

Furthermore, though the classification given has considered disparate sources and outcomes of DDoS attacks, much emphasis has not been laid on the correlation of alerts, which has the tendency to reveal the source of the attack given a substantial amount of instances representing the attack. When flooding attacks are considered in terms of the amount of traffic generated without recourse to the casual relationships between such traffic instances, it can be difficult to trace the attack source in real time.

Preventive measures for cyber-attacks have been discussed in cloud based solutions. One of such approaches proposed in [28] is called SnortFlow, which provides an OpenFlow-based IPS for cloud environments. One significant advantage of this technique is the ability to reconfigure the cloud networking system on-the-fly as a countermeasure when an attack is detected.

To realise the IPS, the authors modeled the system to include such components as the cloud cluster, controller, and OpenFlow switches. The cloud cluster hosts cloud resources in the proposed system and it is based on XenServer, which is a parallel virtualization solution. The resources at different cloud servers are then linked by OpenFlow switches (OFS), and layer-2 connections and other higher technologies are supported at diverse physical cloud servers. Furthermore, the controller stays at the centre of the architecture to exert control or view over the OpenFlow enabled infrastructure.

To enhance the efficiency of the approach, a deviation from the normal process of taking direct action on the suspicious traffic is effected by the reconfiguration of the entire cloud environment to protect it from intrusions. This is achieved with the use of the network programmability feature, which is a component of both the Open vSwitch (OVS) and OpenFlow switch (OFS). The countermeasures that can be applied to protect the network include traffic redirection and isolation, deep packet inspection, change of media access control (MAC) and Internet protocol (IP) addresses, port blocking, and quarantine.

The evaluation of results show that SnortFlow has better performance by inheriting the intrusion detection ability of snort including the highly network reconfigurable nature of OpenFlow. With different deployment scenarios, it is unlikely that SnortFlow will perform well for high speed packets as the experimentation has shown a threshold of 2500 packets for maximum performance and the tendency to drop packets when performing beyond this threshold. Improvements to this model are relevant to ensure its feasibility in real time environments including the use of a multi controller systems for giving instructions to several OFS and OVS simultaneously.

In further research, a software-defined intrusion prevention system given as SDNIPS is proposed for preventing cloud based attacks in [29]. The approach, which includes a detection component, combines Snort-based intrusion detection system and Open vSwitch (OVS). The architecture of SDNIPS is further simplified such that the cloud resources generate the traffic that goes through the SDNIPS agent. The traffic is then matched against the Snort rules and any found match is indicated as an alert that further populates the log file. This alert information is then captured by the SDNIPS daemon and sent to the JSON server at the controller's end. At the processing of the alert information, the alert interpreter parses this information and extracts the required details, which include the attack type, source and destination IP addresses, TCP port, among others.

In subsequent processing of the alert information, the OVS updates the flow table using the OpenFlow rules entries from the rules generator. Any suspicious traffic matching the updated flow table entries is then acted upon by applying the necessary countermeasures in the data plane. In this way, an attack is truncated and the cloud resources can be protected from compromise.

The approach demonstrates an efficient prevention system but the use of Snort creates room for total dependence on expert knowledge for the definition of the rules, which can be time consuming. At the same time, the approach will likely lead to a higher

consumption rate for computing resources since traffic has to pass through the IPS at all times, and delay in matching each traffic pattern against the preset rules will have a significant impact on the resources of the host system.

As the security needs of organisations increase, more and more robust solutions are required to protect the huge resource space. One aspect of achieving a tenable security solution is to have a cost effective, customizable and scalable product or approach. This is the focus of [30] in their proposed real time approach for detecting and preventing attacks.

The approach, which is developed based on the software engineering framework – requirement analysis, design, implementation, and testing configures snort in inline mode to achieve intrusion prevention. Configuring snort in inline mode allows the IPS to deploy its sensors in capturing and dropping suspicious packets, which are likely to have attack payload. The dropped packets are eventually logged with Splunk. This notwithstanding, the inability of signature based intrusion detection and prevention systems such as Snort to detect unknown attacks including its poor performance with heavy network traffic is a major drawback for this approach [11].

5. Conclusion

With several evasive techniques in the public domain, it is relevant in the context of protecting the cyberspace to have in place a robust and stable mechanism for detecting and predicting the likelihood of an attack in a typical network environment. Varying network configurations represent varying activity profiles and behavioural attributes of users and programs. To achieve an effective mechanism in this direction requires a cascade of multiple layers of nonlinear processing components, which can be useful for feature extraction and transformation in order to interpret the dynamic profiles of a network.

A careful evaluation of the proposed approaches mentioned in this paper, in terms of attack detection, prediction, and prevention, shows that dependence on stored signatures and patterns of known attacks as well as tightly coupled behavioral profiles are common. This implies that anticipated attack profiles must match the stored data before an attack can be reported. Though anomaly detection may seem to be a solution in this regard, it is actually a partial solution as it triggers a plethora of false positives, thereby increasing the workload of the security analyst or administrator.

With the evolving threat landscape and sophisticated evasive techniques proliferating, it is imperative to have an approach that is able to learn multiple levels of representations corresponding to different levels of abstractions such that these levels form a cascade of concepts. With these concepts, hidden layers can serve as inputs to the next layer, and in the process, developing feature representations in the internal layers to arrive at a precise or near precise output for detecting and predicting multistage cyberattacks. This implies that an attack protection mechanism can quickly adapt and learn from previous experiences based on a few number of samples to infer and detect the likelihood of novel attacks as well as truncate them in real time.

References

[1] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection”, IEEE Communications Surveys & Tutorials, vol. 18, no. 2, **(2016)**, pp. 1153-1176.
[2] U. Akyazi, “Possible scenarios and maneuvers for cyber operational area”, In European Conference on Cyber Warfare and Security, Academic Conferences International Limited, Greece, **(2014)** July 3-4.
[3] D. E. Denning, “Framework and principles for active cyber defense”, Computers & Security, vol. 40, **(2014)**, pp. 108-113.
[4] N. B. Aissa and M. Guerroumi, “Semi-supervised statistical approach for network anomaly detection”, Procedia Computer Science, vol. 83, **(2016)**, pp. 1090-1095.
[5] G. Kim, S. Lee and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection”, Expert Systems with Applications, vol. 41, no. 4, **(2014)**, pp. 1690-1700.

[6] S. Yoo, S. Kim, A. Choudhary, O. P. Roy and T. Tuithung, "Two-phase malicious web page detection scheme using misuse and anomaly detection", International Journal of Reliable Information and Assurance, vol. 2, no. 1, **(2014)**, pp. 1-9.
[7] M. S. Rani and S. B. Xavier, "A Hybrid Intrusion Detection System Based on C5. 0 Decision Tree Algorithm and One-Class SVM with CFA", International Journal of Innovative Research in Computer, vol. 3, no. 6, **(2015)**, pp. 5526-5537.
[8] W. C. Lin, S. W. Ke and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors", Knowledge-based systems, vol. 78, **(2015)**, pp. 13-21.
[9] H. Shapoorifard and P. Shamsinejad, "A Novel Cluster-based Intrusion Detection Approach Integrating Multiple Learning Techniques", International Journal of Computer Applications, vol. 166, no. 3, **(2017)**, pp. 13-16.
[10] J. Song, H. Takakura, Y. Okabe and K. Nakao, "Toward a more practical unsupervised anomaly detection system", Information Sciences, vol. 231, **(2013)**, pp. 4-14.
[11] A. Abduvaliyev, A. S. K. Pathan, J. Zhou, R. Roman and W. C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks", IEEE Communications Surveys & Tutorials, vol. 15, no. 3, **(2013)**, pp. 1223-1237.
[12] I. Butun, S. D. Morgera and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks", IEEE communications surveys & tutorials, vol. 16, no. 1, **(2014)**, pp. 266-282.
[13] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas and Y. L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system", Information Sciences, vol. 378, **(2017)**, pp. 484-497.
[14] Y. Han, T. Alpcan, J. Chan, C. Leckie and B. I. Rubinstein, "A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing co-residence using semi-supervised learning", IEEE Transactions on information Forensics and Security, vol. 11, no. 3, **(2016)**, pp. 556-570.
[15] M. A. Alsheikh, S. Lin, D. Niyato and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications", IEEE Communications Surveys and Tutorials, vol. 16, no. 4, **(2014)**, pp. 1996-2018.
[16] X. Xu, L. Zuo and Z. Huang, "Reinforcement learning algorithms with function approximation: Recent advances and applications", Information Sciences, vol. 261, **(2014)**, pp. 1-31.
[17] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks", Engineering Applications of Artificial Intelligence, vol. 32, **(2014)**, pp. 228-241.
[18] M. S. Hoque, M. Mukit, M. Bikas and A. Naser, "An implementation of intrusion detection system using genetic algorithm", International Journal of Network Security & Its Applications (IJNSA), vol. 4, no. 2, **(2012)**, pp. 109-120.
[19] I. Ahmad, M. Hussain, A. Alghamdi and A. Alelaiwi, "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components", Neural Computing and Applications, vol. 24, nos. 7-8, **(2014)**, pp. 1671-1682.
[20] S. Rastegari, P. Hingston and C. P. Lam, "Evolving statistical rulesets for network intrusion detection", Applied Soft Computing, vol. 33, **(2015)**, pp. 348-359.
[21] W. Xing-zhu, "Network Intrusion Prediction Model based on RBF Features Classification", International Journal of Security and Its Applications, vol. 10, no. 4, **(2016)**, pp. 241-248.
[22] M. Ghasemi Gol and A. Ghaemi-Bafghi, "E-correlator: an entropy-based alert correlation system", Security and Communication Networks, vol. 8, no. 5, **(2015)**, pp. 822-836.
[23] A. A. Ramaki, M. Amini and R. E. Atani, "RTECA: Real time episode correlation algorithm for multi-step attack scenarios detection", Computers & Security, vol. 49, **(2015)**, pp. 206-219.
[24] U. Ravale, N. Marathe and P. Padiya, "Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function", Procedia Computer Science, vol. 45, **(2015)**, pp. 428-435.
[25] S. Chen, Z. Zuo, Z. P. Huang and X. J. Guo, "A graphical feature generation approach for intrusion detection", In MATEC Web of Conferences, EDP Sciences, vol. 44, **(2016)**.
[26] S. Patil and B. B. Meshram, "Intrusion Prevention System", International Journal of Emerging trends in Engineering and Development, vol. 4, no. 2, **(2012)**, pp. 577-584.
[27] S. T. Zargar, J. Joshi and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks", IEEE Communications Surveys & Tutorials, vol. 15, no. 4, **(2013)**, pp. 2046-2069.
[28] T. Xing, D. Huang, L. Xu, C. J. Chung and P. Khatkar, "Snortflow: A openflow-based intrusion prevention system in cloud environment", In Research and Educational Experiment Workshop (GREE), Second GENI, IEEE, **(2013)**, pp. 89-92.
[29] T. Xing, Z. Xiong, D. Huang and D. Medhi, "SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds", In Network and Service Management (CNSM), 10th International Conference, IEEE, **(2014)**, pp. 308-311.
[30] P. S. Kenkre, A. Pai and L. Colaco, "Real time intrusion detection and prevention system", In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), Springer, Cham, **(2014)**, pp. 405-411.

