Machine Learning Techniques in Cybersecurity

Subjects: Computer Science, Artificial Intelligence

Contributor: Mostofa Ahsan , Kendall E. Nygard , Rahul Gomes , Md Minhaz Chowdhury , Nafiz Rifat , Jayden F Connolly

Machine learning is of rising importance in cybersecurity. The primary objective of applying machine learning in cybersecurity is to make the process of malware detection more actionable, scalable and effective than traditional approaches, which require human intervention. The cybersecurity domain involves machine learning challenges that require efficient methodical and theoretical handling. Several machine learning and statistical methods, such as deep learning, support vector machines and Bayesian classification, among others, have proven effective in mitigating cyber-attacks. The detection of hidden trends and insights from network data and building of a corresponding data-driven machine learning model to prevent these attacks is vital to design intelligent security systems.

cybersecurity machine learning neural networks homomorphic encryption

1. Introduction

Machine learning (ML) is typically described as a branch of “Artificial Intelligence” that is closely related to data mining, computational statistics and analytics and data science, particularly focusing on allowing systems to learn from historical data [1][2]. As a result, machine learning models are often made up of a set of rules, procedures or complex functions and equations. These features can be used to uncover intriguing data patterns, recognize sequences or anticipate behavior [3]. As a result, ML could be useful in the field of cybersecurity.

Machine learning algorithms—in this case, shallow models—are further classified into and supervised learning and unsupervised learning. In supervised learning, the models usually do not have a dependent variable and mostly rely on the internal patterns available in the dataset to group the data into different categories. This can be achieved using different algorithms, such as K-means, Sequential Pattern Mining, DB scan [4] and the a priori algorithm [5]. In supervised learning, the models usually have class labels to verify the predictions. Naïve Bayes, for example, uses probabilistic distribution to identify to which category a class label belongs. Decision trees create a tree-like structure based on a training set. For prediction, once the tree is built, any unknown record can be sorted based on the tree structure. Random forest [6] uses a similar approach, but instead of building one decision tree, it builds multiple decision trees and then uses a voting scheme to classify a record. Because of the collective nature of the decision-making process, random forest usually has higher classification accuracy. Support vector machine (SVM) [7] works by creating a linear decision boundary from the dataset. This can be compared to a binary

classification. SVMs are also capable of transforming the data using a kernel trick. This allows SVMs to classify nonlinear datasets as well.

Deep learning models can also classify or cluster algorithms. However, their approach is quite different from machine learning models. Unlike their counterparts, they do not have a fixed algorithm for prediction. Hence, they are also known as black box models because they analyze the data, identify patterns and use the patterns for production. Deep learning models use artificial neural networks, which are built using several perceptrons. These perceptrons are connected in a randomized fashion during the onset of model training. By looking at the data and training over a given time, the perceptrons gain values, also known as weights, that are better suited for classifying the dataset at hand. There are different varieties of deep learning models. Convolutional neural networks find application in classifying image data. They have also been used to classify cybersecurity datasets by transforming the data into a format that resembles an image. Recurrent neural networks (RNN) find application in classifying data that have a temporal aspect. Several improved versions of RNN include LSTM or long short-term memory, as well as Bi-LSTM. Unsupervised learning with deep learning includes autoencoders as well as generative adversarial networks. Autoencoders mostly use feature reduction, where information is transformed into a compressed form before further processing. They are well adapted to compressing information in a meaningful way, thereby increasing the prediction accuracy.

Reinforcement learning explores a different approach of training a model to differentiate between long-term and short-term goals. Agents interact with their environment and, based on their actions, they are rewarded or penalized. The reward is variable, thereby teaching the model to become better. A popular example is DQN [8] or Deep Q Networks. In DQN, the mapping function between states and actions is accomplished using deep learning, thereby reducing the requirement for a large table of Q-learning (TQL). A variant of DQN is QR-DQN [9], which uses quantile regression to model the possible distributions, instead of returning the mean distribution. This can be compared to the difference between decision trees and random forest, summarized above.

2. Stages of a Cyber-Attack

Organizations can assess the cybersecurity risk to them and can identify certain security threats. They can then implement security controls or measures against these threats. They can utilize the National Institute of Standards and Technology (NIST) Special Publications, although they may not be a US federal agency or related contractor [10]. NIST Special Publications provide step-by-step guidance for applying a risk management framework to federal information systems. In this guidance, a set of security issues are identified and common controls or measures against these security issues/threats are listed. In a recent study, machine learning tools were suggested as efficient controls or measures [11]. Such measures can be applicable to all five phases of a cyber-attack.

There are five phases of a cyber-attack. They are reconnaissance, scan, attack (denial-of-service attacks, gain access using application and operating system attacks, network attacks), maintain access (using Trojans, backdoors, rootkits, etc.) and cover tracks and hiding. An interruption at any phase can either interrupt or halt the

entire process of attack. Machine learning algorithms can be used in all of these phases to help fight against cyber-attacks by disrupting the attacker's workflow.

During the reconnaissance or preparation phase of the attack, an adversary uses techniques such as a social engineering attack (phishing, malicious call, etc.). Machine learning algorithms can look for email signatures and detect malicious or phishing email signatures and block them. There are cases when an attacker calls the target organization and impersonates a third party to obtain valuable information (known as "voice phishing" or "vishing"). Call source analysis using machine learning algorithms can flag and block such calls. Another example use of machine learning is scanning any external devices connected to the organization's property, e.g., a USB device. Such a scan prevents malicious software from propagating through such devices. Another example is when the adversary wishes to guess the access password to obtain unauthorized access (violating confidentiality) [12]. Rule-based machine learning algorithms can detect the most common passwords that are used by the organization's employees and can recommend a list of unrecommended passwords. This will hinder the reconnaissance step. Such machine learning algorithms can be placed in strategic locations, e.g., key machines and networks.

During the scan phase, sometimes called "Weaponization", the cyber-attacker or adversary exploits the vulnerabilities of the target system. The attacker uses automated tools, such as Metasploit, AutoSploit and Fuzzers [12]. Machine learning algorithms can be used to automatically scan and find the vulnerabilities by an ethical hacker before the adversary can. For example, a machine learning-based penetration test can be implemented, specifically by integrating the algorithms into the penetration testing tools, e.g., Metasploit. Such algorithms, upon being used by a pen tester, can find novel weaknesses.

Machine learning algorithms are a strong measure against the attacks (phase 3 of cyber-attack). Machine learning algorithms that can be used to provide cybersecurity are linear regression, polynomial regression, logistic regression, naïve Bayes classifier, support vector machine, decision tree, nearest neighbor, clustering, dimensionality reduction, linear discriminant analysis and boosting [13]. The applications of these algorithms as a measure against cybersecurity problems are spam detection (includes phishing), malware detection, denial-of-service attacks (including DDoS) and network anomaly detection. Other forms of attacks are associated with biometric recognition, authentication, identity theft detection and social media analytics. Information leakage detection, APT or advanced persistent threat detection, hidden channel detection and software vulnerability detection are also some modern threats that need addressing.

During phase four of a cyber-attack, malware is used to maintain access by the attacker, e.g., Trojans, backdoors or rootkits. Machine learning algorithms can detect such malware traffic packets when the malware contacts the attacker and vice versa. For example, for malware detection, support vector machines (SVM) are an efficient option [14]. SVM was implemented using Weka to detect Android OS malware (260 samples), using static features analysis. Here, a black box method was used by analyzing the malware's behavior rather than executing the malware. In the first step, a Python code was used to extract Android application packages' (APK) features, one package at a time. Both malicious (201 samples) and benign (59 samples) APKs were selected. In the second step, an SVM classifier (Weka and LibSVM classifier) was trained by these features, to identify malware from these

APKs. In the testing phase, the used APKs were downloaded from the repositories: Android, Malware Dataset, Kaggle, Drebin, Android Malshare and APKPure. The receiver operating characteristic or ROC curve was used to present the result. An enhancement of this application used the dynamic features of malware, as malware keeps changing its features. An SVM model can be trained to perform binary classification from a set of features of network packets. The trained classifier can detect a DDoS attack by identifying normal vs. abnormal network traffic, especially for IoT devices. Examples of the features used to train machine learning algorithms include the destination IP address, sequence number and minimum, maximum and average packets for each destination IP address, received signal strength indication, network allocation vector, value injection rate and inter-arrival time between consecutive frames, etc. The traffic information was collected by placing sensors at significant points of the network, e.g., at the gateway level, for a traffic session of 15–20 min. This classifier can be used as an extra security layer in IDS.

Another example is the application of various clustering techniques (K-means, DBSCAN and Hierarchical) [14]. Clustering is useful for malware detection, phishing attack detection, spam filtering and detecting the larger family of software bugs known as side-channel attack detection. In [14], both malware and goodware Android APKs were installed in an Android emulator. Then, their resource usage statistics (CPU and RAM) were recorded for all three clustering techniques. For all three clustering algorithms, a total of 217 data instances were used, with 145 for training and 72 for testing. The conclusion was that CPU-RAM usage statistics are not an efficient feature for clustering malware and goodware. The nearest neighborhood (NN) search is used in access control. For example, an NN is used to identify actual vs. forged biometrics (e.g., fingerprints) through classification based on their patterns [14]. The CSD200 model was used as the fingerprint scanner to take 100 samples (10 people, total 100 fingers). MATLAB was used to convert these images into an array or matrix. Such a machine learning algorithm can automatically make decisions as to whether a biometric is forged or original. In [14], decision trees were used, e.g., Iterative dichotomizer 3 (ID 3) and its successor, C4.5, to identify malware efficiently. The dataset used was from the Cardiff University Research Portal. In another research work [15], anomalous services running into the computer systems, both offline and online, were identified using a neural network-based model (NARX-RNN), AI-based multi-perspective SVM, principal component analysis (PCA) and hierarchical process tree-based reinforcement learning techniques.

During phase five or the covering tracks phase, the attacker wishes to confirm that their identification is not being tracked. They employ different techniques, including corrupting machine learning tools' training data to misidentify their data. The machine learning algorithms themselves can be robust but their training data may not be. Deceptive training data make the algorithm inefficient. This process of forging training data is called adversarial machine learning (AML). The severity is serious for cybersecurity applications. The countermeasures against such polluted data include game theory (non-cooperative game/Nash equilibrium, Zero-Sum Versus Non-Zero Sum Game, simultaneous move vs. sequential game or Bayesian Game) [16]. An example of AML is network traffic classification. Performing deep packet inspection is hard when the traffic payload is encrypted [17]. For such traffic, it is possible that the machine learning classifier (e.g., network scanning detector) is deceived by an adversary, to tag malware or botnet communications or NMap network scanning traffic as benign. It is possible that the adversary can mimic the features of benign traffic and can infer the classification output. What happens if the

adversary's traffic is classified as malicious? The adversary does not obtain any feedback but their traffic will probably be blocked. This will give them an indication that their traffic has been classified as malicious and prompt the adversary to change the traffic signature. Improved machine learning techniques exist that can work as a measure against adversarial attacks [18]. For example, an activation clustering method was introduced that identifies the hidden layer of a deep neural network where an adversarial trigger lies. Using empirical learning algorithms, poisonous data points can be identified when poisoning attacks happen against an SVM.

3. Supervised Learning

Supervised learning relies on useful information in historical labeled data. Supervised learning is performed when targets are predefined to reach from a certain set of inputs, i.e., task-driven approach. Classification and regression methods are the most popular supervised learning techniques [19]. These methods are well known for classifying or predicting the target variable for a particular security threat. For example, classification techniques can be used in cybersecurity to indicate a denial-of-service (DoS) attack (yes, no) or identify distinct labels of network risks, such as scanning and spoofing. Naive Bayes [20], support vector machines (SVM) [21], decision tree [22][23], K-nearest neighbors [24], adaptive boosting [25] and logistic regression [26] are some of the most well-known classification techniques in shallow models.

Naive Bayes finds a good amount of use in cybersecurity. The researchers in [27] used the naive Bayes classifier from the Weka package and KDD'99 data for training and testing. Data were grouped into the four attack types (probe and scan, DoS, U2R and R2L) and their classifier achieved 96%, 99%, 90% and 90% testing accuracy, respectively. The cumulative false positive rate was 3%. The researchers in [28] developed a framework using a simple form of Bayesian network using the KDD'99 data and used categories to depict different attack scenarios. Solving an anomaly detection problem, the reported results were 97%, 96%, 9%, 12% and 88% accuracy for normal, DoS, R2L, U2R and probe or scan categories, respectively. The false positive rate was not reported but can be inferred to be less than 3%. Naive Bayes was also used as one of the methods in [29] to solve a DoS problem, which attempted to resolve the botnet traffic in filtered Internet Relay Chat (IRC), therefore determining the botnet's existence and origin. The study conducted used TCP-level data that were collected from 18 different locations on the Dartmouth University campus' wireless network. This data collection occurred over a span of four months. A filter layer was used to extract IRC data from the network data. Labeling was a challenge so the study utilized simulated data for the experiments. The performance of the Bayesian network showed 93% precision with a false positive rate of 1.39%. C4.5 decision trees were also used for comparison and achieved 97% precision, but the false positive rates were higher, at 1.47% and 8.05%, respectively.

In [30], the researchers used an SVM classifier to detect DDoS attacks in a software-defined network. Experiments were conducted on the DARPA dataset, comparing the SVM classifier with other standard classification techniques. Although the classifier had higher accuracy, the SVM took more time, which is an obvious flaw. In [31], the researchers used a least-squares SVM to decrease the training time on large datasets. Using three different feature extraction algorithms, they reduced the number of features from 41 to 19. The data were resampled to have around 7000 instances for each of the five classes of the KDD'99 dataset. Overall, the classification was reported

at 99% for DoS, probe or scan, R2L and normal classes and 93% for U2R. Research in [32] utilized a robust SVM, which is a variation of SVM where the discriminating hyperplane is averaged to be smoother and the regularization parameter is automatically determined. Preprocessing training and testing of data was done on the Basic Security Module from the DARPA 1998 dataset. It showed 75% accuracy with no false positives and 100% accuracy with 3% false positives.

In [33], the researchers utilized decision trees to generate detection rules against denial-of-service and command injection attacks on robotic vehicles. Results showed that different attacks had different impacts on robotic behavior. A decision tree-based intrusion detection system that may change after intrusion by analyzing the behavior data through a decision tree was implemented in [34]. The model was used to prevent advanced persistent threat (APT) attacks, which use social engineering to launch various forms of intrusion attacks. The detection accuracy in their experiments was 84.7%, which is very high for this experiment. Decision trees were also used in [35], where the researchers replaced the misuse detection engine of SNORT with decision trees. SNORT is a known tool that follows the signature-based approach. The researchers utilized clustering of rules and then derived a decision tree using a version of an ID3 algorithm. This rule clustering reduced the number of comparisons to determine which rules were triggered by given input data, and the decision tree located the most discriminating features of the data, thereby achieving parallel feature evaluation. This method achieved superior performance to that of SNORT when applied to the 1999 DARPA intrusion detection dataset. The results varied drastically depending on traffic type. The fastest were up 105%, with an average of 40.4% and minimum of 5% above the normal detection speed of SNORT. The number of rules was also increased from 150 to 1581, which resulted in a pronounced speed-up versus SNORT.

Ensemble learning techniques such as random forest (RF) have also been explored in cybersecurity research. Random forest uses multiple decision trees to draw a conclusion and can be more accurate than a single decision tree. In [36], the researchers employed a random forest algorithm on the KDD dataset for misuse, anomaly and hybrid-network-based intrusion detection. Patterns were created by the random forest and matched with the network for misuse detection. To detect anomalies, the random forest detected novel intrusions via outliers. Using the patterns built by the model, new outliers were also discovered. The study implemented a full system solution including anomaly detection. Data were grouped into majority attacks and minority attacks. This hybrid system achieved superior performance for misuse detection, where the error rate on the original dataset was 1.92%, and it was 0.05% for the balanced dataset.

Regression algorithms are useful for forecasting a continuous target variable or numeric values, such as total phishing attacks over a period of time or network packet properties. In addition to detecting the core causes of cybercrime, regression analysis can be utilized for various risk analysis forms [37]. Linear regression [1], support vector regression [21] and random forest regressor are some of the popular regression techniques. The main distinction between classification and regression is that, in regression, the output variable is numerical or continuous, but in classification, the projected output is categorical or discrete. Ensemble learning is an extension of supervised learning that mixes different shallow models, e.g., XGBoost and random forest learning [6], to complete a specific security task.

4. Unsupervised Learning

The main goal of unsupervised learning, also known as data-driven learning, is to uncover patterns, structures or relevant information in unlabeled data [38]. Risks such as malware can be disguised in a variety of ways in cybersecurity, including changing their behavior dynamically to escape detection. Clustering techniques, which are a form of unsupervised learning, can aid in the discovery of hidden patterns and insights in datasets, as well as the detection of indicators of sophisticated attacks. In addition, clustering algorithms can effectively spot anomalies and policy violations, recognizing and eliminating noisy occurrences in data, for example. K-means [39] and K-medoids [40] are clustering algorithms used for partitioning, while single linkage [41] and complete linkage [42] are some of the most widely utilized hierarchical clustering methods in various application sectors. Furthermore, well-known dimensionality reduction techniques such as linear discriminant analysis (LDA), principal component analysis (PCA) and Pearson correlation, as well as positive matrix factorization, can be used to handle such problems [1]. Another example is association rule mining, in which machine learning-based policy rules can learn to avert cyber incidents. The rules and logic of an expert system are normally manually documented and implemented by a knowledge engineer working with a domain expert [38][43][44]. In contrast, association rule learning identifies the rules or associations among a set of security aspects or variables in a dataset [45]. Different statistical analyses are performed to quantify the strength of associations [37]. In the domain of machine learning and data mining, various association rule mining methods have been presented, such as tree-based [46], logic-based [47], frequent pattern-based [48][49][50], etc. Moreover, a priori [48], a priori-TID and a priori-Hybrid [48], AIS [45], FP-Tree [46], RARM [51] and Eclat [52] have been used extensively for association rule learning. These algorithms are able to resolve such difficulties by creating a set of cybersecurity policy rules.

The researchers in [53] applied sequential pattern mining on the DARPA 1999 and 2000 data to reduce the redundancy of alerts and minimize false positives. Using the a priori algorithm, they discovered attack patterns and created a pattern visualization for the users with a support threshold of 40%. They were able to detect 93% of the attacks in twenty seconds. The study reported a real-time scenario with 84% attack detection, with the main variation coming from the support threshold. In [54], DBSCAN was used as a clustering method to group normal versus anomalous network packets in the KDD'99 dataset. The dataset was preprocessed and features were selected using correlation analysis. The performance was shown at 98% for attack or no-attack detection. The researchers in [55] took a different data mining approach to create an intrusion detection system called ADMIT. This system does not rely on labeled data; rather, it builds user profiles incrementally using a dynamic clustering technique. It uses sequences, which are a list of tokens collected from users' data, to build a profile. These sequences are classified as normal or anomalous based on using a modified form of K-means clustering. They set a value of 20 for the maximum sequence length, which resulted in 80% performance accuracy with a 15% false positive rate. A unique algorithm based on the signature a priori algorithm was used to find new signatures of attacks from existing attack signatures, proposed in [56]. Here, the researchers compared their algorithm processing time to that of the a priori and found that their algorithm was much faster.

5. Artificial Neural Networks (ANN)

ANN is a segment of machine learning, in the area of Artificial Intelligence, that is a computationally complex model inspired by the biological neural networks in the human brain [1]. The basic idea behind ANN was first introduced in the 1940s, with ANNs becoming a popular idea and technology in the late 1970s and continuing into the 1990s. Although SVMs were prominent in the 1990s, overshadowing the ANNs, they have received popularity recently and are steadily increasing in use. ANNs consist of multiple neurons that work together in layers to extract information from the dataset. The primary difference is the performance of ANN versus shallow machine learning as the amount of security data grows. The number of studies of ANN-based intrusion detection systems has increased rapidly from 2015 to the present. In [57], the researchers utilized ANNs to detect misuse. Using data generated by a RealSecure network monitor, ten thousand events were collected, of which 3000 were simulated attacks by programs. Preprocessing of the data was performed and ten percent of the data were selected randomly for testing, with the rest used for training the ANN. The error rates for training and testing were 0.058 and 0.070, respectively. Each packet was categorized as normal or attack. A combination of keyword selection and ANN was proposed in [58] to improve IDS. Keyword selection was performed on Telnet sessions and statistics were derived from the number of times that the keywords occurred (from a predefined list). These statistics were given as input to the ANN and the output identified the probability of an attack. The researchers in [59] compared fuzzy logic and artificial neural networks to develop comprehensive intrusion detection systems and tested them using the KDD'99 dataset. They presented a detailed discussion on the preprocessing, training and validation of the proposed approach. The "class" characteristic in this dataset, which is made up of around 5 million data instances with 42 properties, determines whether a particular instance is a typical connection instance or one of the four types of attacks that need to be recognized. Five different machine learning approaches were compared, among which the FC-ANN-based approach [60] and the hierarchical SOM-based approach [61] were the best detectors.

Deep learning models, which are a form of ANN, learn feature representations directly from the original data, such as photos and texts, without the need for extensive feature engineering. As a result, deep learning algorithms are more effective and need less data processing. For large datasets, deep learning methods have a significant advantage over classical machine learning models. Some of the widely used deep learning techniques in cybersecurity include deep belief networks (DBNs), convolutional neural networks (CNNs) and recurrent neural networks (RNNs) as supervised learning models. Several variants of autoencoders, restricted Boltzmann machines (RBMs) and generative adversarial networks (GANs) have been used with success for unsupervised learning. The researchers in [62] used DBNs to detect malware with an accuracy of 96.1%. The DBNs used unsupervised learning to discover layers of features and then used a feed-forward neural network to optimize discrimination. DBNs can learn from unlabeled data, so, in the experiments, DBNs provided a better classification result than SVM, KNN and decision tree. DBNs were also used in [63] to create an IDS. The proposed four-layer model was used on the KDD'99 Cup dataset, where the researchers reported accuracy, precision and false acceptance rate (FAR) of 93.49%, 92.33% and 0.76%, respectively. In [64], a DBN-based ad hoc network intrusion detection model was developed with an experiment on the Network Simulator (NS2) platform. The experiment showed that this method can be added to the ad hoc network intrusion detection technology. Accuracy and FAR were reported as 97.6% and 0.9%, respectively. A deep learning approach called DeepFlow was proposed in [65] to directly detect malware in Android applications. The architecture consisted of three components for feature extraction, feature

coarse granularity and classification. Two modules were used to assess malware sources from the Google Play Store. Experiments showed that DeepFlow outperformed SVM, ML-based algorithms and multi-layer perceptron (MLP).

References

1. Han, J.; Kamber, M.; Pei, J. Data mining concepts and techniques third edition. Morgan Kaufmann Ser. Data Manag. Syst. 2011, 5, 83–124.
2. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. Practical machine learning tools and techniques. Morgan Kaufmann 2005, 2, 578.
3. Dua, S.; Du, X. Data Mining and Machine Learning in Cybersecurity; CRC Press: Boca Raton, FL, USA, 2016.
4. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD-94, Oregon, Portland, 2–4 August 1996; Volume 96, pp. 226–231.
5. Inokuchi, A.; Washio, T.; Motoda, H. An apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, 13–16 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 13–23.
6. Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.
7. Cortes, C.; Vapnik, V. Support-vector networks. Mach. Learn. 1995, 20, 273–297.
8. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. arXiv 2013, arXiv:1312.5602.
9. Dabney, W.; Rowland, M.; Bellemare, M.; Munos, R. Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
10. Force, J.T. Risk management framework for information systems and organizations. NIST Spec. Publ. 2018, 800, 37.
11. Breier, J.; Baldwin, A.; Balinsky, H.; Liu, Y. Risk Management Framework for Machine Learning Security. arXiv 2020, arXiv:2012.04884.
12. Buchanan, B.; Bansemer, J.; Cary, D.; Lucas, J.; Musser, M. Automating Cyber Attacks: Hype and Reality; Center for Security and Emerging Technology: Washington, DC, USA, 2020.

13. Alazab, M.; Venkatraman, S.; Watters, P.; Alazab, M. Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011.
14. Thomas, T.; Vijayaraghavan, A.P.; Emmanuel, S. Machine Learning Approaches in Cyber Security Analytics; Springer: Berlin/Heidelberg, Germany, 2020.
15. Sakthivel, R.K.; Nagasubramanian, G.; Al-Turjman, F.; Sankayya, M. Core-level cybersecurity assurance using cloud-based adaptive machine learning techniques for manufacturing industry. Trans. Emerg. Telecommun. Technol. 2020, 33, e3947.
16. Dasgupta, P.; Collins, J. A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks. AI Mag. 2019, 40, 31–43.
17. De Lucia, M.J.; Cotton, C. Adversarial machine learning for cyber security. J. Inf. Syst. Appl. Res. 2019, 12, 26.
18. Xi, B. Adversarial machine learning for cybersecurity and computer vision: Current developments and challenges. Wiley Interdiscip. Rev. Comput. Stat. 2020, 12, e1511.
19. Sarker, I.H.; Kayes, A.; Watters, P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. J. Big Data 2019, 6, 1–28.
20. John, G.H.; Langley, P. Estimating continuous distributions in Bayesian classifiers. arXiv 2013, arXiv:1302.4964.
21. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Comput. 2001, 13, 637–649.
22. Salzberg, S.L. C4. 5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc. Mach. Learn. 1994, 16, 235–240.
23. Sarker, I.H.; Colman, A.; Han, J.; Khan, A.I.; Abushark, Y.B.; Salah, K. Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model. Mob. Netw. Appl. 2020, 25, 1151–1161.
24. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-based learning algorithms. Mach. Learn. 1991, 6, 37–66.
25. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. ICML 1996, 96, 148–156.
26. Le Cessie, S.; Van Houwelingen, J.C. Ridge estimators in logistic regression. J. R. Stat. Soc. Ser. Appl. Stat. 1992, 41, 191–201.
27. Panda, M.; Patra, M.R. Network intrusion detection using naive bayes. Int. J. Comput. Sci. Netw. Secur. 2007, 7, 258–263.

28. Amor, N.B.; Benferhat, S.; Elouedi, Z. Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2004; pp. 420–424.
29. Carl, L. Using machine learning technliques to identify botnet traffic. In Proceedings of the 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 14–16 November 2006; IEEE: Piscataway, NJ, USA, 2006.
30. Kokila, R.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 205–210.
31. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. J. Netw. Comput. Appl. 2011, 34, 1184–1199.
32. Hu, W.; Liao, Y.; Vemuri, V.R. Robust Support Vector Machines for Anomaly Detection in Computer Security. In Proceedings of the ICMLA, Los Angeles, CA, USA, 23–24 June 2003; pp. 168–174.
33. Vuong, T.P.; Loukas, G.; Gan, D.; Bezemskij, A. Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In Proceedings of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
34. Moon, D.; Im, H.; Kim, I.; Park, J.H. DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. J. Supercomput. 2017, 73, 2881–2895.
35. Kruegel, C.; Toth, T. Using decision trees to improve signature-based intrusion detection. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 8–10 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 173–191.
36. Zhang, J.; Zulkernine, M.; Haque, A. Random-forests-based network intrusion detection systems. IEEE Trans. Syst. Man Cybern. Part Appl. Rev. 2008, 38, 649–659.
37. Watters, P.A.; McCombie, S.; Layton, R.; Pieprzyk, J. Characterising and predicting cyber attacks using the Cyber Attacker Model Profile (CAMP). J. Money Laund. Control 2012, 15, 430–441.
38. Sarker, I.H. Context-aware rule learning from smartphone data: Survey, challenges and future directions. J. Big Data 2019, 6, 1–25.
39. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June–18 July 1965; Volume 1, pp. 281–297.

40. Ricci, F.; Rokach, L.; Shapira, B. Introduction to recommender systems handbook. In Recommender Systems Handbook; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–35.
41. Sneath, P.H. The application of computers to taxonomy. Microbiology 1957, 17, 201–226.
42. Sorensen, T.A. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol. Skar. 1948, 5, 1–34.
43. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity 2019, 2, 1–22.
44. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst. Appl. 2014, 41, 1690–1700.
45. Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 26–28 May 1993; pp. 207–216.
46. Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. ACM Sigmod Rec. 2000, 29, 1–12.
47. Flach, P.A.; Lachiche, N. Confirmation-guided discovery of first-order rules with Tertius. Mach. Learn. 2001, 42, 61–95.
48. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference Very Large Data Bases, VLDB, Santiago, Chile, 12–15 September 1994; Volume 1215, pp. 487–499.
49. Houtsma, M.; Swami, A. Set-oriented mining for association rules in relational databases. In Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 6–10 March 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 25–33.
50. Liu, B.; Hsu, W.; Ma, Y. Integrating classification and association rule mining. Knowl. Discov. Data Min. Inf. 1998, 98, 80–86.
51. Das, A.; Ng, W.K.; Woon, Y.K. Rapid association rule mining. In Proceedings of the Tenth International Conference on Information and Knowledge Management, Atlanta, GA, USA, 5–10 October 2001; pp. 474–481.
52. Zaki, M.J. Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. 2000, 12, 372–390.
53. Li, Z.; Zhang, A.; Lei, J.; Wang, L. Real-time correlation of network security alerts. In Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE’07), Hong Kong, China, 24–26 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 73–80.

54. Blowers, M.; Williams, J. Machine learning applied to cyber operations. In Network Science and Cybersecurity; Springer: Berlin/Heidelberg, Germany, 2014; pp. 155–175.
55. Sequeira, K.; Zaki, M. Admit: Anomaly-based data mining for intrusions. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 386–395.
56. Zhengbing, H.; Zhitang, L.; Junqi, W. A novel network intrusion detection system (nids) based on signatures search of data mining. In Proceedings of the First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), Adelaide, Australia, 23–24 January 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 10–16.
57. Cannady, J. Artificial neural networks for misuse detection. In Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, USA, 5–8 October 1998; pp. 443–456.
58. Lippmann, R.P.; Cunningham, R.K. Improving intrusion detection performance using keyword selection and neural networks. Comput. Netw. 2000, 34, 597–603.
59. Li, J.; Qu, Y.; Chao, F.; Shum, H.P.; Ho, E.S.; Yang, L. Machine learning algorithms for network intrusion detection. In AI in Cybersecurity; Springer: Berlin/Heidelberg, Germany, 2019; pp. 151–179.
60. Wang, G.; Hao, J.; Ma, J.; Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst. Appl. 2010, 37, 6225–6232.
61. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. A hierarchical SOM-based intrusion detection system. Eng. Appl. Artif. Intell. 2007, 20, 439–451.
62. Ding, Y.; Chen, S.; Xu, J. Application of deep belief networks for opcode based malware detection. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3901–3908.
63. Gao, N.; Gao, L.; Gao, Q.; Wang, H. An intrusion detection model based on deep belief networks. In Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 20–22 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 247–252.
64. Tan, Q.S.; Huang, W.; Li, Q. An intrusion detection method based on DBN in ad hoc networks. In Proceedings of the International Conference on Wireless Communication and Sensor Network (WCSN 2015), Changsha, China, 12–13 December 2015; World Scientific: Singapore, 2016; pp. 477–485.
65. Zhu, D.; Jin, H.; Yang, Y.; Wu, D.; Chen, W. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In Proceedings of the 2017

IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 438–443.