A Review on Artificial Intelligence Methods For Cyber Intrusion Detection

[1]Vaishnavi Ganesh
Assistant Professor
[1]Computer Science And Engineering
[1]Priyadarshini Indira Gandhi College of Engineering, Nagpur, India

Abstract : This survey paper describes a focused literature survey of machine learning (ML) and data mining (DM) methods for cyber analytics in support of intrusion detection.Based on the number of citations or the relevance of an emerging method, papers representing each method were identified, read, and summarized. The discussion of challenges for using ML/DM for cyber security is presented, and some recommendations on when to use a given method are provided.

IndexTerms **- Cyber analytics, data mining, machine learning.**

I. INTRODUCTION

Cyber security is the set of technologies and processes designed to protect computers, networks, programs, and data from attack, unauthorized access, change, or destruction. Cyber security systems are composed of network security systems and computer (host) security systems. Each of these has, at a minimum, a firewall, antivirus software, and an intrusion detection system (IDS). IDSs help discover, determine, and identify unauthorized use, duplication, alteration, and destruction of information systems [1]. The security breaches include external intrusions (attacks from outside the organization) and internal intrusions (attacks from within the organization)[2].

There are three main types of cyber analytics in sup-port of IDSs: misuse-based (sometimes also called signature-based), anomaly-based, and hybrid. Misuse-based techniquesare designed to detect known attacks by using signatures of those attacks. They are effective for detecting known type of attacks without generating an overwhelming number of falsealarms. They require frequent manual updates of the database with rules and signatures. Misuse-based techniques cannot detect novel (zero-day) attacks.

Anomaly-based techniques model the normal network and system behavior, and identify anomalies as deviations from normal behavior. They are appealing because of their ability to detect zero-day attacks. Another advantage is that the profiles of normal activity are customized for every system, application, or network, thereby making it difficult for attackers to know which activities they can carry out undetected. Additionally, the data on which anomaly-based techniques alert (novel attacks) can be used to define the signatures for misuse detectors. The main disadvantage of anomaly-based techniques is the potential for high false alarm rates (FARs) because previously unseen (yet legitimate) system behaviors may be categorized as anomalies.

Hybrid techniques combine misuse and anomaly detection.They are employed to raise detection rates of known intrusions and decrease the false positive (FP) rate for unknown attacks.An in-depth review of the literature did not discover many pure anomaly detection methods; most of the methods were really hybrid. Therefore, in the descriptions of ML and DM methods, the anomaly detection and hybrid methods are described together.

Another division of IDSs is based on where they look for intrusive behavior: network-based or host-based. A network-based IDS identifies intrusions by monitoring traffic through network devices. A host-based IDS monitors process and file activities related to the software environment associated with a specific host.

II. RELATED WORK

Nguyen et al.[3] describe ML techniques for Internet traffic classification. The techniques described therein do not rely on well-known port numbers but on statistical traffic characteristics. Unlike Nguyen et al. [3], this paper presents methods that work on any type of cyber data, not only Internet Protocol (IP) flows.

Teodoro et al.[4] focus on anomaly-based network intrusion techniques. The authors present statistical, knowledge-based, and machine-learning approaches, but their study does not present a full set of state-of-the-art machine-learning methods. In contrast, this paper describes not only anomaly detection but also signature-based methods. Our paper also includes the methods for recognition of type of the attack (misuse) and for detection of an attack (intrusion). Lastly, our paper presents the full and latest list of ML/DM methods that are applied to cyber security.

Sperotto et al.[5] focus on Network Flow (NetFlow) data and point out that the packet processing may not be possible at the streaming speeds due to the amount of traffic. They describe a broad set of methods to detect anomalous traffic (possible attack) and misuse. However, unlike our paper, they do not include explanations of the technical details of the individual methods.

Wu et al. [6] focus on Computational Intelligence methods and their applications to intrusion detection. Methods such as Artificial Neural Networks (ANNs), Fuzzy Systems, Evolutionary Computation, Artificial Immune Systems, and Swarm Intelligence are described in great detail. Because only Computational Intelligence methods are described, major ML/DM methods such as clustering, decision trees, and rule mining (that this paper addresses) are not included.

This paper focuses primarily on cyber intrusion detection as it applies to wired networks. With a wired network, an adversary must pass through several layers of defense at firewalls and operating systems, or gain physical access to the network.However, a wireless network can be targeted at any node, so it is naturally more vulnerable to malicious attacks than a wired network. The ML and DM methods covered in this paper are fully applicable to the intrusion and misuse detection problems in both wired and wireless networks.

III. MAJOR STEPS IN ML AND DM

ML focuses on classification and prediction,based on known properties previously learned from the training data. ML algorithms need a goal (problem formulation) from the domain (e.g., dependent variable to predict). DM focuses on the discovery of previously unknown properties in the data. It does not need a specific goal from the domain, but instead focuses on finding new and interesting knowledge.

One can view ML as the older sibling of DM.

There are three main types of ML/DM approaches: unsupervised, semi-supervised, and supervised. In unsupervised learning problems, the main task is to find patterns, structures, or knowledge in unlabeled data. When a portion of the data is labeled during acquisition of the data or by human experts, the problem is called semi-supervised learning. The addition of the labeled data greatly helps to solve the problem. If the data are completely labeled, the problem is called supervised learning and generally the task is to find a function or model that explains the data.

Once a classification model is developed by using training and validation data, the model can be stored so that it can be used later or on a different system. The Predictive Model Markup Language (PMML) is developed and proposed by Data Mining Group to help predictive model sharing [7].It is based on XML and currently supports logistic regression and feed-forward neural network (NN) classifiers. The latest version supports Naïve Bayes, k-Nearest Neighbor (k-NN), and Support Vector Machine (SVM) classifiers. The model supports several common DM metadata such as a data dictionary (e.g., discrete, Boolean, numerical), normalization, model name, model attributes, mining schema, outlier treatment, and output.

IV. ML AND DM METHODS FOR CYBER SECURITY

This section describes the different ML/DM methods for cyber security. Each technique is described with some detail, and references to seminal works are provided.

A. Artificial Neural Networks

ANNs are inspired by the brain and composed of interconnected artificial neurons capable of certain computations on their inputs [8]. The input data activate the neurons in the first layer of the network whose output is the input to the second layer of neurons in the network. Similarly, each layer passes its output to the next layer and the last layer outputs the result. Layers in between the input and output layers are referred to as hidden layers. When an ANN is used as a classifier, the output layer generates the final classification category.

Bivens et al. [9] describe a complete IDS that employs a preprocessing stage, clustering the normal traffic, normalization, an ANN training stage, and an ANN decision stage. The first stage used a Self-Organizing Map (SOM), which is a type of unsupervised ANN, to learn the normal traffic patterns over time, such as commonly used TCP/IP port numbers. In this manner, the first stage quantized the input features into bins, which were then fed to the second stage, a Multilayer Perceptron (MLP) ANN. The MLP network parameters, such as the number of nodes and layers, were determined by the first stage SOM. Once the MLP training was completed, it started predicting the intrusions. The system can be restarted for a new SOM to learn a new traffic pattern and a new MLP attack classifier to be trained.

B. Association Rules and Fuzzy Association Rules

An association rule describes a relationship among different attributes: IF (A AND B) THEN C. This rule describes the relationship that when A and B are present, C is present as well. Association rules have metrics that tell how often a given relationship occurs in the data. The support is the prior probability (of A, B, and C), and the confidence is the conditional probability of C given A and B.

A simple example of an association rule pertaining to the items that people buy together is:

IF (Bread AND Butter) → Milk (1)

This rule states that if a person buys bread and butter, they also buy milk.

A limitation of traditional Association Rule Mining is that it only works on binary data [i.e., an item was either purchased in a transaction (1) or not (0)]. In many real-world applications, data are either categorical (e.g., IP name, type of public health intervention) or quantitative (e.g., duration, number of failed logins, temperature). For numerical and categorical attributes,

Boolean rules are unsatisfactory. An extension that can process numerical and categorical variables is called Fuzzy Association Rule Mining [10].

Fuzzy association rules are of the form:

IF (X is A) → (Y is B) (2)

where X and Y are variables, and A and B are fuzzy sets that characterize X and Y, respectively. A simple example of fuzzy association rule for a medical application could be the following:

IF (Temperature is Strong Fever) AND (Skin is Yellowish) AND (Loss of appetite is Profound) → (Hepatitis is Acute)

The rule states that if a person has a Strong Fever, Yellowish skin and Profound Loss of appetite, then the person has Acute Hepatitis. Strong Fever, Yellowish, Profound, and Acute are membership functions of the variables Temperature, Skin, Loss of appetite, and Hepatitis, respectively.

C. Bayesian Network

A Bayesian network is a probabilistic graphical model that represents the variables and the relationships between them [11], [12]. The network is constructed with nodes as the discrete or continuous random variables and directed edges as the relationships between them, establishing a directed acyclic graph. The child nodes are dependent on their parents. Each node maintains the states of the random variable and the conditional probability form. Bayesian networks are built using expert knowledge or using efficient algorithms that perform inference.

Each state (or network variable) can be an input to other states with certain set of state values. For example, the protocol state can pick values from available protocol numbers. Each of the state values that can go from a state to another state have an associated probability, and the sum of those probabilities will add up to 1 representing the entire set of state values. Depending on the application, the network can be used to explain the interplay between the variables or to calculate a probable outcome for a target state (e.g., alert or file access) using the input states.

D. Clustering

Clustering [13] is a set of techniques for finding patterns in high-dimensional unlabeled data. It is an unsupervised pattern discovery approach where the data are grouped together based on a similarity measure. The main advantage of clustering for intrusion detection is that it can learn from audit data without requiring the system administrator to provide explicit descriptions of various attack classes.

There are several approaches for clustering the input data. In connectivity models (e.g., hierarchical clustering), data points are grouped by the distances between them. In centroid models (e.g., k-means), each cluster is represented by its mean vector. In distribution models (e.g., Expectation Maximization algorithm), the groups are assumed to be acquiescent to a statistical distribution. Density models group the data points as dense and connected regions (e.g., Density-Based Spatial Clustering of Applications with Noise [DBSCAN]). Lastly, graph models (e.g., clique) define each cluster as a set of connected nodes (data points) where each node has an edge to at least one other node in the set.

An instance-based learning (also called lazy learning) algorithm, the k-NN, is another popular ML method where the classification of a point is determined by the k nearest neighbors of that data point.

E. Decision Trees

A decision tree is a tree-like structure that has leaves, which represent classifications and branches, which in turn represent the conjunctions of features that lead to those classifications. An exemplar is labeled (classified) by testing its feature (attribute) values against the nodes of the decision tree. The best known methods for automatically building decision trees are the ID3 [14] and C4.5 [15] algorithms. Both algorithms build decision trees from a set of training data using the concept of information entropy. When building the decision tree, at each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of examples into subsets. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then performs recursion on the smaller subsets until all the training examples have been classified.

The advantages of decision trees are intuitive knowledge expression, high classification accuracy, and simple implementation. The main disadvantage is that for data including categorical variables with a different number of levels, information gain values are biased in favor of features with more levels.

F. Ensemble Learning

In general, supervised learning algorithms search the hypothesis space to determine the right hypothesis that will make good predictions for a given problem. Although good hypotheses might exist, it may be hard to find one. Ensemble methods combine multiple hypotheses, hoping to form a better one than the best hypothesis alone. Often, ensemble methods use multiple weak learners to build a strong learner [16].

A weak learner is one that consistently generates better predictions than random. One of the ensemble algorithms uses boosting to train several weak learning algorithms and combine (i.e., summation) their weighted results. Adaptive Boosting (AdaBoost) [17] is one of the more popular algorithms used to reduce the over-fitting problem inherent to ML. Boosting can be

seen as a linear regression where data features are the input to the weak learner h (e.g., a straight line dividing the input data points into two categories in space), and the output of the boosting is the weighted summation of these h functions.

Bagging (bootstrap aggregating) is a method to improve the generality of the predictive model to reduce over-fitting. It is based on a model-averaging technique and known to improve the 1-nearest neighbor clustering performance.

The Random Forest classifier [18] is an ML method that combines the decision trees and ensemble learning. The forest is composed of many trees that use randomly picked data features (attributes) as their input. The forest generation process constructs a collection of trees with controlled variance.The resulting prediction can be decided by majority voting or weighted voting.

G. Evolutionary Computation

The term evolutionary computation encompasses Genetic Algorithms (GA) [19], Genetic Programming (GP) [20], Evolution Strategies [21], Particle Swarm Optimization [22], Ant Colony Optimization [23], and Artificial Immune Systems [24]. This subsection focuses on the two most widely used evolutionary computation methods—GA and GP. They are both based on the principles of survival of the fittest. They operate on a population of individuals (chromosomes) that are evolved using certain operators. The basic operators are selection, crossover, and mutation. They start usually with a randomly generated population. For each individual from the population, a fitness value is computed that reveals how good a given individual is at solving the problem at hand. The individuals with higher fitness have a higher probability of being chosen into the mating pool and thus being able to reproduce. Two individuals from the mating pool can perform crossover (i.e.,exchange genetic material between them) and each can also undergo mutation, which is a random alteration of the genetic material of the individual. The highest fit individuals are copied into the next generation.

H. Hidden Markov Models

Markov chains and Hidden Markov Models (HMMs) belong to the category of Markov models. A Markov chain [25] is a set of states interconnected through transition probabilities that determine the topology of the model. An HMM [26] is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters. The main challenge is to determine the hidden parameters from the observable parameters. The states of an HMM represent unobservable conditions being modeled. By having different output probability distributions in each state and allowing the system to change states over time, the model is capable of representing non-stationary sequences.

I. Support Vector Machine

The SVM is a classifier based on finding a separating hyperplane in the feature space between two classes in such a way that the distance between the hyperplane and the closest data points of each class is maximized. The approach is based on a minimized classification risk [27] rather than on optimal classification. SVMs are well known for their generalization ability and are particularly useful when the number of features, m, is high and the number of data points, n, is low (m >> n).

When the two classes are not separable, slack variables are added and a cost parameter is assigned for the overlapping data points. The maximum margin and the place of the hyperplane is determined by a quadratic optimization with a practical runtime of O(n2), placing the SVM among fast algorithms even when the number of attributes is high.

V. COMPARISON CRITERIA

There are several criteria by which the ML/DM methods for cyber could be compared:

- Accuracy
- Time for training a model
- Time for classifying an unknown instance with a trained model
- Understandability of the final solution (classification)

If one were to compare the accuracy of several ML/DM methods, those methods should be trained on exactly the same training data and tested on exactly the same testing data. Unfortunately, even in the studies that used the same data set (e.g., KDD 1999), when they compared their results with the best methods from the KDD Cup (and usually claimed their results were better), they did so in an imperfect fashion—they used a subset of the KDD data set, but not necessarily the same subset that the other method used. Therefore, the accuracy of these results is not comparable.

The time for training a model is an important factor due to ever changing cyber-attack types and features. Even anomaly detectors need to be trained frequently, perhaps incrementally, with fresh malware signature updates.

Time for classifying a new instance is an important factor that reflects the reaction time and the packet processing power of the intrusion detection system.

Understandability or readability of the classification model is a means to help the administrators examine the model features easily in order to patch their systems more quickly. This information (such as packet type, port number, or some other high level network packet feature that reflects the cyber-attack footpath) will be available through the feature vectors that are tagged by the classifier as an intrusion category.[2]

VI. CONCLUSION

The paper describes the literature review of ML and DM methods used for cyber. Special emphasis was placed on finding example papers that describe the use of different ML and DM techniques in the cyber domain, both for misuse and anomaly detection. Unfortunately, the methods that are the most effective for cyber applications have not been established; and given the richness and complexity of the methods, it is impossible to make one recommendation for each method, based on the type of attack the system is supposed to detect. When determining the effectiveness of the methods, there is not one criterion but several criteria that need to be taken into account. They include accuracy, complexity, time for classifying an unknown instance with a trained model, and understandability of the final solution (classification) of each ML or DM method. Depending on the particular IDS, some might be more important than others.

Another crucial aspect of ML and DM for cyber intrusion detection is the importance of the data sets for training and testing the systems. ML and DM methods cannot work without representative data, and it is difficult and time consuming to obtain such data sets. To be able to perform anomaly detection and misuse detection, it is advantageous for an IDS to be able to reach network- and kernel-level data.

They are especially related to how often the model needs to be retrained. A fertile area of research would be to investigate the methods of fast incremental learning that could be used for daily updates of models for misuse and anomaly detection.

REFERENCES

[1] A. Mukkamala, A. Sung, and A. Abraham, "Cyber security challenges:Designing efficient intrusion detection systems and antivirus tools," in Enhancing Computer Security with Smart Technology, V. R. Vemuri, Ed.New York, NY, USA: Auerbach, 2005, pp. 125–163.

[2] Anna L.Buczak, Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection",in IEEE COMMUNICATIONS SURVEYS & TUTORIALS,VOL.18,NO 2,SECOND QUARTER 2016.

[3] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," IEEE Commun. Surv.Tuts., vol. 10, no. 4, pp. 56–76, Fourth Quart. 2008.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Comput. Secur., vol. 28, no. 1, pp. 18–28, 2009.

[5] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," IEEE Commun.Surv. Tuts., vol. 12, no. 3, pp. 343–356, Third Quart. 2010.

[6] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," Appl. Soft Comput., vol. 10, no. 1, pp. 1–35, 2010.

[7] A. Guazzelli, M. Zeller, W. Chen, and G. Williams, "PMML an open standard for sharing models," R J., vol. 1, no. 1, pp. 60–65, May 2009.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Netw., vol. 2, pp. 359 366,1989.

[9] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, and M. Embrechts, "Network-based intrusion detection using neural networks," Intell. Eng. Syst. Artif. Neural Netw., vol. 12, no. 1, pp. 579–584, 2002.

[10] C. M. Kuok, A. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," ACM SIGMOD Rec., vol. 27, no. 1, pp. 41–46, 1998.

[11] D. Heckerman, A Tutorial on Learning with Bayesian Networks. New York, NY, USA: Springer, 1998.

[12] F. V. Jensen, Bayesian Networks and Decision Graphs. New York, NY, USA: Springer, 2001.

[13] K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[14] R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986.

[15] R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[16] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits Syst. Mag., vol. 6, no. 3, pp. 21–45, Third Quart. 2006.

[17] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in Proc. 13th Int. Conf. Mach. Learn., 1996, vol. 96, pp. 148–156.

[18] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[19] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," Mach. Learn., vol. 3, no. 2, pp. 95–99, 1988.

[20] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA, USA: MIT Press, 1992.

[21] H. G. Beyer and H. P. Schwefel, "Evolution strategies: A comprehensive introduction," J. Nat. Comput., vol. 1, no. 1, pp. 3–52, 2002.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., 1995, vol. IV, pp. 1942–1948.

[23] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 53–66, Apr. 1997.

[24] J. Farmer, N. Packard, and A. Perelson, "The immune system, adap- tation and machine learning," Phys. D: Nonlinear Phenom., vol. 2, pp. 187–204, 1986.

[25] A. Markov, "Extension of the limit theorems of probability theory to a sum of variables connected in a chain," Dynamic Probabilistic Systems, vol. 1, R. Howard. Hoboken, NJ, USA: Wiley, 1971 (Reprinted in Appendix B).

[26] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," Bull. Amer. Math. Soc., vol. 73, no. 3, p. 360, 1967.

[27] V. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: Springer, 2010.