ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 10(35), DOI: 10.17485/ijst/2017/v10i35/118951, September 2017* ISSN (Online) : 0974-5645

Survey of Data Mining (DM) and Machine Learning (ML) Methods on Cyber Security

Rahul D. Shanbhogue and B. M. Beena

Department of CSE, New Horizon College of Engineering, Marathalli, Bangalore – 560103, India;
rahuldshanbhogue@gmail.com, beena.nh@gmail.com

Abstract

Objectives: This paper is survey on how the Machine Learning (ML) and Data-Mining (DM) techniques have been employed to automate the cyber detection system and discusses necessary background knowledge on Cyber Security. **Findings:** After identifying various issues on cyber intrusion detection and security, also various Machine Language and Data Mining approaches that have been employed to resolve this. **Analysis:** This paper sheds light on complexities, peculiarities and potential of using ML Algorithms for Cyber Security. **Application:** The machine learning and data mining algorithms and procedures discussed below are applied in cyber security intrusion detection systems in real time scenarios.

Keywords: Anomaly Detection, Data-Mining (DM), Intrusion Detection System (IDS), Machine Learning (ML), Misuse Detection

1. Introduction

Cyber security is set of technologies and processes designed to protect systems in a network from external and internal attacks, unauthorised access or destruction. A cyber security system consists of two main parts a network security system and host security system, both with a minimum of firewalls, antivirus software and Intrusion Detection System (IDS). IDS help identify unauthorised use, alteration, duplication, and destruction of information systems[1].

There are three types of cyber analytics in support for IDS – Misuse based, Anomaly based, Hybrid based. Misuse detectors detect attacks based in known signatures and require frequent updates. They cannot detect zero day or novel attacks but generate least false rate. Anomaly detectors, model network and system behaviour and identify deviations from normal behaviour. Capable to detect novel attacks and can be used to define signatures for misuse detectors. This method has potentially high false alarm rates. Hybrid detectors combine misuse and anomaly detection and are employed to increase the detection rates and decrease false positive rate of unknown attacks.

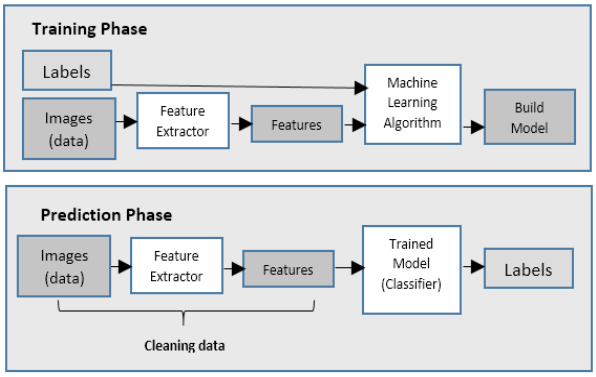

Figure 1. ML phases.

1.1 Major Steps in ML

ML is a data analysis method that automates building of an analytical model using algorithms that learn from data which can be easily automated, and find insights in the data without being explicit programming as to where

*Author for correspondence

to look. Machine Language is a computer program that learns from Experience (E) with respect to some class of Task (T) and Performance measure (P). If its performance with task 'T' as measured by 'P' improves with 'E'.

ML has three phases – training, validation, and testing. To decide which best model of the alternatives is, the selection should be based on the performance of the model against validation data and not on the accuracy on test data set. The following steps are performed:

1. Identify the features from training data.
2. Identify subset of attributes necessary for classification (dimensional reduction).
3. Learn model using training data.
4. Use trained model to classify unknown data, and predict the result accurately.

1.1.1 Supervised Knowledge Induction

The construction of new knowledge has been called inductive or empirical learning, because it relies heavily on data, i.e., specific experiences or objects, to produce hypothesis that generalize the data. The hypothesis produced in this manner is therefore implicitly attended by varying degrees of uncertainty. Since the label can be thought of as being predetermined and provided to the learning system by an entity (the 'supervisor'), learning from labelled objects has been called *supervised learning*.

1.1.2 Regression Supervised Learning

Predict from function of continuous valued output.

Figure 2. Regression learning.

1.1.3 Classification Supervised Learning

Predict from function of discrete valued output.

A labelled instance can be viewed as a pair (x; f(x)); where x is the instance itself and the function f(x) returns its label. The goal of supervised inductive learning is therefore to compute a function f' that approximates f, which, in turn, defines the target concept.

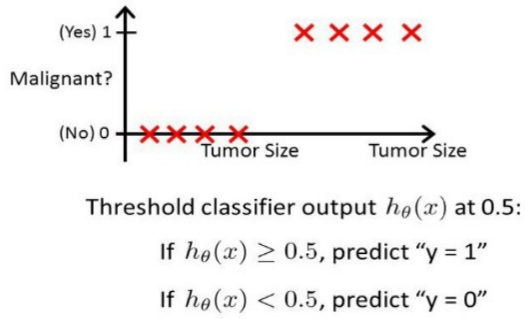

Figure 3. Classification learning.

Moreover, f' can be learned *incrementally* (involves taking an instance from a training set and revising the current hypothesis or hypothesis set(s) so that it is consistent with this instance. This process normally continues until all instances in the training set have been processed. Alternatively, f' can be learned *non-incrementally* (involves examining the training set, selecting a sub-set of instances from the training set, and revising a hypothesis or hypothesis set so that it covers (i.e., is satisfied by) this subset of instances. This process normally continues until no more instances remain to be covered.

When training the model sometimes model may result into the below 2 scenarios:

1. *Under-fitting,* which is result of excessively simple model.
2. *Over-fitting,* which is result of excessively complicated model.

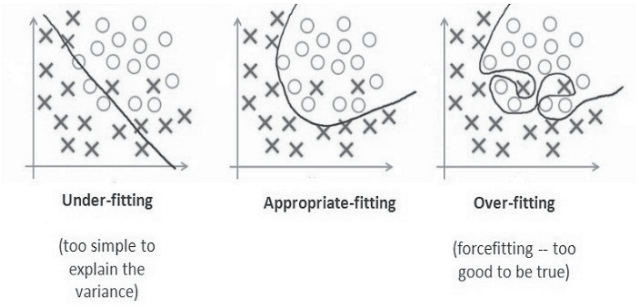

Figure 4. Graph plotted to suggest the kinds fitting of a hypothesis function.

Artificial Neural Networks (ANNs) are inspired by the brain and composed of interconnected artificial neurons capable of certain computations on their inputs[2]. 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. ANN classifiers are based on the perceptron[3] and were very popular until the 1990s when SVMs were invented.

Genetic or Evolutionary Algorithms use a set of 'genetic patterns'. Each genetic pattern denotes an individual and is usually represented as a string of bits. Moreover, each pattern has an associated fitness value that summarizes its past performance. Learning using genetic algorithms involves updating the fitness values of the individuals, performing operations such as crossover (gene splicing) and mutation on the fitter individuals to produce new ones, and using these new individuals to replace those which are less fit. Genetic algorithms are a primary learning component of so called classifier systems.

Neural and Evolutionary Learning approaches are composed of nodes called units connected by weighted, directed links. Each unit has a current activation level, a set of input links from other units, functions for computing the unit's next activation level given its inputs and their weights, and a set of output links. Inputs to a network typically come in the form of a vector of Boolean features. The output of a network is the activation of the designated output unit(s). Learning is accomplished by making small adjustments in the weights of the links using some rule, to reduce the error/mean squared error between the observed and predicted output values, thereby making the network consistent with the examples.

Bayesian Networks are probabilistic directed acyclic graphs used for reasoning under uncertainty. By ensuring that each node (which denotes a random variable) in the graph is conditionally independent of its predecessors given its parents, Bayesian networks allow a more convenient representation and manipulation of the entire joint probability distribution of a domain. Specifically, one need only specify the prior probabilities of the root nodes and the conditional probabilities of non-root nodes given their immediate predecessors. This also allows us to view a link in the graph as representing direct dependency or causality. When the network structure is known and all variables are observable, learning in Bayesian networks reduces to estimating, from statistics of the data, the conditional probabilities of the networks' links. However, when not all variables can be readily observed, learning becomes strikingly like that which occurs in feed forward neural networks.

Unsupervised Knowledge Induction: Most ML systems learn from labelled instances, nevertheless it is also possible to learn from unlabelled objects but difficult. Such an approach is called *unsupervised learning.* They are algorithms used against data that has no historical classifications. The system is not aware of the «right answer.» In this algorithm, we do not have any target or outcome variable to predict. The goal is to explore the data and find some structure within. There is no given hypothesis function f to approximate and measure against. The most popular approach of generalising unlabelled instances is *conceptual clustering,* where clustering is the task of grouping a set of objects in the same group are more like each other.

Reinforcement Learning: machine is trained to make decisive actions. The machine is exposed to an environment where it trains itself indefinitely using trial and error, and learns which actions yield the best rewards. This machine learns from experience and tries to capture the best possible knowledge to make most accurate decisions (used typically for robotics, gaming and navigation).

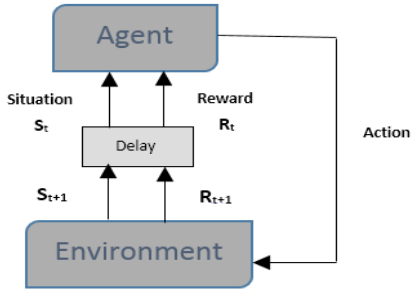

Figure 5. Overview of working of reinforced learning.

Decision tree is a graphical representation. It makes use of branching methodologies which illustrates all possible outcomes of a decision, based on specific conditions. In a decision tree, the internal node represents a test on the attribute, each branch of the tree represents the outcome of the test and the leaf node represents a class label

i.e. the decision made after computing all the attributes. The classification rules are illustrated through the path formed from the edges connecting root to the leaf node.

An exemplar is classified by testing its attribute values against the nodes of the decision tree. While building the decision tree, at each node of the tree, choose the attribute of the data that most effectively splits its dataset into subsets. The splitting measure is the normalized information gain. The feature with the highest normalized information gain is picked to settle on the decision.

The algorithm then recursively splits the current subset into smaller subsets until all the training examples have been labelled. The benefit of employing decision trees is high classification accuracy, and simple implementation. The prominent drawback is that for data including categorical variables with different number of levels, information gain values are biased in favour of features with more levels. The decision tree is built by maximizing the information gain at each variable split, resulting in a natural variable ranking or feature selection.

Therefore, we have identified the main approaches and paradigms of ML techniques and briefly sketched each above.

1.2 Major Steps in DM

Knowledge Discovery in Databases (KDD) full process dealt with extracting useful previously unknown information from data using DM techniques to apply specific algorithms to extract patterns from data.

Misuse class is learned by using appropriate exemplars from training set and new data is run on the model and the exemplar is classified to one of the misuse class. If exemplar doesn't belong to any misuse class its labelled normal. In Anomaly detection, network traffic is defined in training phase, learned model is applied to new data and every exemplar is classified normal or anomalous.

DM is the process of examining large pre-existing databases to discover and generate actionable information. DM uses mathematical analysis to derive patterns that exist in data. Typically, relationships between the data are too complex and the datasets are too humongous that these patterns cannot be discovered by traditional data exploration but can be collected and defined by a DM model.

This process can be defined by using the following six basic steps:

1.2.1 Business Understanding (Defining the Problem)

The first step entails analysing requirements, defining the extent of the subject matter, defining the metrics via which the model will be evaluated and defining specific objectives. These can be translated into following questions: What is the data you're looking for? What relationships are seen among data? Which feature are we predicting? What actions need to be performed (such as cleansing, aggregation, or processing) to make the data usable? To answer these queries, you need to conduct a data availability study, to find the needs of the business, and its users.

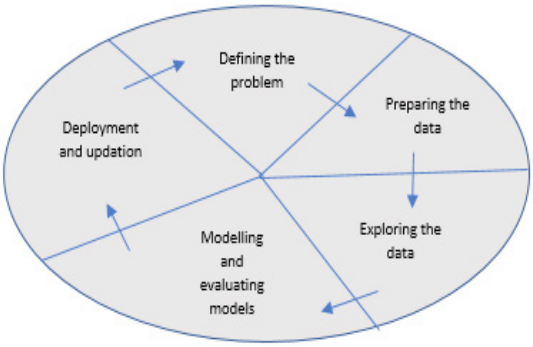

Figure 6. Overview of data mining phases.

1.2.2 Data Preparation

This step performs cleaning of data that was identified in the former step. Data can be scattered as well as be stored in different formats, and contain inconsistencies. Cleansing of Data is not just about removing inaccurate data or interpolating missing values, but is finding insights in the data and recognise sources that are the most accurate, and selecting the most appropriate for use.

1.2.3 Data Exploration

By exploring the data, you can decide if the dataset contains flawed data, and then you can devise a strategy for gaining a deeper understanding. It involves techniques such as calculating the minimum and maximum values, calculating mean and standard deviations, and looking at the distribution of the data. They provide useful information about the stability and accuracy of the results. A large standard deviation suggests that adding more data might help improve the model.

1.2.4 Modelling

This step focusses on building the mining model or models. You will make use of the knowledge that you gained in the former step to help you construct the models. Here model is processed which is often referred to as training which refers to the process of applying a specific algorithm to the defined data in the structure to discover patterns.

1.2.5 Evaluating and Validating the Model

This step involves evaluating mining models that you have built and test their performance. Before a model is deployed into the environment, its best practice to test how well the model performs. Analysis Services provides tools that separate your data into training and testing datasets so that you can accurately assess the performance of all models on the same data. Also, when you build a model, you typically construct multiple alternate models with different configurations to see which yields the best results for the defined problem and dataset.

1.2.6 Deployment and Updation of Model

The last step is to deploy the model that yields the best predictions, which help you take best business decisions. Update the models after review and analysis which involves reprocessing the model to improve the effectiveness of the solution.

The process illustrated is cyclic, that is a DM model is a dynamic and iterative process, each step in the process might need to be reiterated to create a good model.

2. Literature Survey

Several studies use KDD data sets as they were easy to obtain and contained network and OS level data. Most important is the type and level of the input data.

The attack data coming to network stack, and effects of packets on OS level carrying important information. Hence, it's important that IDS can reach network and kernel level data.Type of ML and DM algorithms selected and overall structure of system.

Packet-Level Data: There are 144 IPs listed by the Internet Engineering Task Force (IETF). User programs running the widely used protocols (such as, TCP, UDP, ICMP, etc.) generate the packet network traffic of the Internet. The network packets received and transmitted at the physical interface (e.g., Ethernet port) is captured by a specific application programming interface (API) called pcap. It contains libraries for many network tools, including protocol analysers, packet sniffers, network monitors, and network IDSs, and traffic generators.

NetFlow Data: NetFlow was popularised as a router feature by Cisco. The router can collect IP network traffic as it enters and leaves the interface. Cisco's NetFlow version 5 defines a network flow as a unidirectional sequence of packets that shares the exact same seven packet attributes: ingress interface, source IP address, destination IP address, IP protocol, source port, destination port, and IP type of service. The NetFlow data include a compressed and pre-processed version of the actual network packets.

Misuse Detection: Misuse Detection, classifies abnormal network traffic based on Clustering methods (particularly density clustering algorithms), since they are:

1. versatile,
2. easy to implement,
3. less parameterized,
4. high processing speeds.

The work mentioned in[4] an SVM classifier was used to classify the KDD 1999 dataset into predefined categories (DoS, Probe or Scan, and Normal). From the 41 features, a subset of attributes was selected by following a feature removal policy and feature selection policy. The work also focused on a subset of the training set which was determined by Ant Colony Optimization Approach, which helped to maximize labelling and minimize the bias in the KDD set. The study reported its validation performance with overall 98% accuracy.

The work mentioned in[5] used a least-squared SVM to have a faster system to train on large data sets which helped reduce the number of attributes in the KDD data set from 41-19. They employed three different feature selection algorithms. The first one, was based on picking the feature that maximizes the classification performance, the second was based on mutual information (proven to be slightly more promising), and the third was correlation based.

SVM performs well, learns from extracting Association rules and Sequential pattern from available normal traffic data.

Anomaly Detection and Hybrid Detection: This classifies attack pattern against known signatures or extracts new signatures from attack labelled data coming from anomaly detection module.

Generates readable signatures, capturable through:

1. Branch features of decision trees
2. Genes in genetic algorithm
3. Association rules or sequential pattern in DM.

Network data cannot be properly modelled by simple distribution - single packet payload may contain data associative to dozens of networks protocol and user behaviour. Methods like Bayesian or HMM mayn't be strongest since data may not have properties appropriate to them.

Evolutionary computation takes long time to run and hence not suitable for real time cases such as training online systems. If attack signature is emphasized, Decision tree, Evolutionary Computation association rules of DM can be employed. Designers should investigate of data of good quality and exploitable statistical properties.

The work mentioned in[6] used NetFlow data collected from real world and simulated attack data using the Flame tool[7] and other ISP sources for real world attack data. The study used one-class SVM classifier, which is considered a natural approach for anomaly detection. A new window kernel was introduced to help find an anomaly based on time position of the NetFlow data where more than one NetFlow record entered this kernel. The performance was reported as 89% to 94% accuracy on the various attack types.

2.1 Computational Complexity of Ml and DM Methods

Factors that determine performance of ML and DM methods in cyber security are as below:

1. Accuracy
2. Time for training a model
3. Time for classifying unknown
4. instance of trained model
5. Readability of final solution

Table 1. Complexity of Ml Anddm algorithms during training

Algorithm	Typical Time Complexity	Streaming Capable
ANN	O(cmnk)	Low
Association Rules	>>O(n^3)	Low
Bayesian Network	>>O(mn)	High
Clustering, k-means	O(kmni)	High
Clustering, hierarchical	O(n^3)	Low
Clustering, DBSCAN	O(n log n)	High
Decision Trees	O(mn^2)	medium
GA	O(gkmn)	Medium
Naïve Bayes	O(mn)	high
Nearest Neighbor k-NN	O(n log k)	High
HMM	O(nc^2)	Medium
Random Forest	O(Mmn log n)	medium
Sequence Mining	>>O(n^3)	Low
SVMs	O(n^2)	medium

Table1 below illustrates the complexity of various ML and DM techniques that are under discussion. As a thumb rule:

1. O(n) and O(n logn) algorithms, have linear time and are suitable for online systems.
2. O(n^2) algorithms acceptable time complexity for most practice.
3. O(n^3) algorithms are suitable for offline systems.

The training time of a model is most distinguishing factor due to ever changing cyber-attack Even anomaly detectors need to be trained frequently, perhaps incrementally, with fresh malware signature updates. This time factor reflects the reaction time and the packet processing time of the intrusion detection system.

Label	Page	Section	Reference
1	1	II.I	SELF
2	2	II.I	https://www.coursera.org/learn/machine-learning/
3	2	II.I	
4	2	II.I	http://vitalflux.com/machine-learning-diagnose-underfittingoverfitting-learning-algorithm/
5	3	II.I	SELF
6	4	II.II	SELF

Table

Label	Page	Section	Refer
1	6	III.I	A Survey on Cyber Security Intrusion Detection Anna L. Buczak, Member, IEEE, and Erhan Guven, Member, IEEE.

2.2 Pecularities of DM and ML

Related to how often model needs to be retrained and availability of labelled data. Unlike in DM and ML, Cyber security training time of model is of highest significance. The model requires to be are trained daily, where the new attacks are identified and pattern becomes known and retraining starts. Area of research is to investigate methods of fast learning incremental approaches for daily retraining of model. In cyber domain, data is harvested from the sensors on the network, to get net flow or TCP.

Complexity lies in the sheer volume of the data and labelling them. Using new datasets help advances in the ML and DM methods for cyber security and worthwhile investments into labelling data since availability of labelled data is scarce. Employing new data set, momentous improvements could be made to ML and DM methods in cyber security and breakthroughs could be possible.

Nevertheless, the best possible available data set right now is the KDD 1999 intrusion detection data set. However, being 15 years old, this data set does not have examples of all the new attacks that have occurred in the last 15 years.

3. Conclusion

The present work defines the essential elements of Cyber Security system modules, and identified several important ML and DM applications on Cyber Intrusion Detection. Further paper also discusses about the approaches and paradigms of ML and briefly sketched each. The study also examined the various ways in which ML techniques have been used in the induction of Cyber Security. Although there is no best approach, Support Vector Machine Algorithms, Genetic and Evolutionary Algorithms, Association rules or sequential pattern in DM. seem to be the most promising approaches for the IDS.

4. References

1. Mukkamala A, Sung A, Abraham A. 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. p. 125–63.
2. Hornik K, Stinchcombe M, White H. Multilayer feed forward networks are universal approximators, Neural Netw. 1989; 2:359–66. Crossref.
3. Rosenblatt F. The perception: A probabilistic model for information storage and organization in the brain, Psychol. Rev. 1958; 65(6):386–408. Crossref. PMid:13602029.
4. Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K. An efficient intrusion detection system based on support vector machines and gradually feature removal method, Expert Syst. Appl. 2012; 39(1):424–30. Crossref.
5. Amiri F, Mahdi M, Yousefi R, Lucas C, Shakery A, Yazdani N. Mutual information-based feature selection for IDSs, J. Netw. Comput. Appl. 2011; 34(4):1184–199. Crossref.
6. Wagner C, Jérôme F, Thomas E. Machine learning approach for IP-flow record anomaly detection. In: Networking 2011.New York, NY, USA: Springer; 2011. 28–39. PMCid:PMC3120781.
7. Brauckhoff D, Wagner A, May M. Flame: A low-level anomaly modeling engine. In: Proc. Conf. Cyber Secur. Exp. Test; 2008.