IDS Using Machine Learning -Current State of Art and Future Directions

Article · March 2016

DOI: 10.9734/BJAST/2016/23668

3 authors:

Yasir Hamid
Abu Dhabi Polytechnic
33 PUBLICATIONS **319** CITATIONS
SEE PROFILE

Sugumaran Muthukumarasamy
Pondicherry Engineering College
37 PUBLICATIONS **298** CITATIONS
SEE PROFILE

Balasaraswathi Ranganathan
SRM Institute of Science and Technology
12 PUBLICATIONS **179** CITATIONS
SEE PROFILE

British Journal of Applied Science & Technology
15(3): 1-22, 2016, Article no.BJAST.23668
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

IDS Using Machine Learning - Current State of Art and Future Directions

Yasir Hamid[1*], M. Sugumaran[1] and V. R. Balasaraswathi[1]

[1]Department of Computer Science and Engineering, Pondicherry Engineering College, India.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/23668
Editor(s):
(1) Wei Wu, Applied Mathematics Department, Dalian University of Technology, China.
Reviewers:
(1) Kexin Zhao, University of Florida, USA.
(2) Valasani Usha Shree, Vidya Jyothi of Institute of Technology (AUTONOMOUS), Hyderabad, India.
Complete Peer review History: http://sciencedomain.org/review-history/13804

Review Article

Received 14th December 2015
Accepted 13th January 2016
Published 21st March 2016

ABSTRACT

The prosperity of technology worldwide has made the concerns of security tend to increase rapidly. The enormous usage of Internetworking has raised the need of protecting systems as well as networks from the unauthorized access or intrusion. An intrusion is an activity of breaking into the system by compromising the security policies, and the process of analyzing the network data for the possible intrusions is Intrusion Detection. For the last two decades automatic intrusion detection system has been an important research topic. Up to the moment, researchers have developed Intrusion Detection Systems (IDS) capable of detecting attacks in several available environments. A boundlessness of methods for misuse detection as well as anomaly detection has been applied, most popular of the all is using machine learning techniques. In this work a survey of various research efforts spared towards the development of intrusion detection systems based on machine learning techniques in given. The surveyed works are presented in easy to understand tabular forms and for each work; technique employed, dataset used and the parameters evaluated are mentioned. Current achievements and limitations in developing intrusion detection system by machine learning and future directions for research are also given.

Keywords: Anomaly; IDS; intrusion; machine learning.

**Corresponding author: E-mail: bhatyasirhamid@pec.edu, vrbala80@gmail.com, sugu@pec.edu;*

1. INTRODUCTION

Being an essential part of daily life and an essential tool today, Internet aids people in diverse areas like business, education, entertainment etc. For the business operations both business and customers apply the Internet applications for business activities [1]. But with the popularity of Internet comes the risk of network attacks or intrusions and the need to secure network against such attacks. Intrusion, an attack on the confidentiality, availability, and integrity is a series of activities aiming at compromising the security of a computer network system [2] taking many forms: external attacks, internal misuses, network-based attacks, information gathering, Denial of Service, and so on [3]. No system can be made perfectly secure because of financial and complexity constraints, hence the hacker will eventually find a way to break into our system, to analyze the network data for the possible intrusions (attacks) an IDS has become the essential component of computer security to supplement existing defenses. Conventional intrusion prevention strategies like access control schemes, firewall or encryption methods have failed to prove themselves to effectively protect networks and systems from increasingly sophisticated attacks and malwares. The Intrusion Detection System (IDS) have become the proper salvage and have become crucial component of any security infrastructure to detect the threats before they cause widespread damage. An IDS is hardware, software, policy or their combination responsible for uncovering the possible intrusions from the network audit data. What makes IDS different from intrusion prevention system (IPS) is that IPS is proactive in nature and tries to prevent an intrusion to occur in network whereas IDS is reactive in nature and works on assumption that no matter how secure a network is intrusions are bound to take place and it tries to uncover if there were really any.

An attacker follows a well-defined ordered series of steps to break into the system and starts with gathering information about the system like the protocol used and the systems available on the network. Once the list of the systems on the network is available, the attacker starts to probe each of the system to list out various vulnerabilities in the system, applications running and the ports open. After the vulnerability is pointed out and the target system is marked the attacker tries to gain the initial access to the target system by performing Remote and Local (R2L) attack. Once the hacker gains a user access on the system, he tries to escalate the privilege he has on the system by performing User to Root (U2R) attacks. After getting the super user privilege on the system the attacker carries out the attack by stealing or modifies confidential or valuable information, modifying web pages, or implanting a backdoor as a stepping stone for future attack purpose, etc. Once a target is compromised, the attacker can do anything he wishes at this stage.

To counter the problem of network attacks a lot of devices have been developed over the last few decades some proactive and some reactive in nature. IDS can be classified as either host based or the network by their defensive scope [4]. Host based IDS captures and analyzes the data on the attacked system itself where the network detection captures and inspects the packets at the network gateway before the attack can reach to the end system [5]. Network based IDS is installed as the second line of defense behind the firewall to protect the LAN. It is aimed at detecting the intrusions caused by multiple hosts. Whereas the host bases system needs to be installed on every machine which makes them efficient for detecting U2R and R2L attacks but at high operation and maintenance cost [6,7]. Both host based and the network have different monitoring domain and both of them detect different attacks effectively.

The rest of the paper is structured as follows: Section 2 gives an overview of the detection techniques employed in IDS. In section 3 various supervised and unsupervised machine learning techniques used in the surveyed works for IDS are discussed. Feature selection techniques are discussed in section 4. Dataset and the tools available are discussed in sections 5. Section 6 discusses about the performance parameters used to check the effectiveness of the works surveyed. Various problems pertaining to current IDS and their possible solutions as well as the future research directions have been discussed in section 7. In section 8 works in IDS for machine learning have been given in tables. Finally section 9 concludes this paper.

2. DETECTION TECHNIQUES

An intrusion detection system (IDS) rounds around the assumption that user behavior is observable and normal user behavior is different from intrusive behavior [8]. At the heart of intrusion detection lies the ability to distinguish acceptable, normal system behavior from that which is abnormal (possibly indicating unauthorized activities) or actively harmful [9]. Two approaches to this problem can be distinguished, with some IDS implementing a combination of both approaches.

2.1 Anomaly Detection

An anomaly detection model attempts to model normal behavior. This technique observes the user behavior over the period of time and builds the model that closely represents user's legitimate (normal) behavior. Events which are very different from this model are considered to be suspicious. For example, a normally passive public web server attempting to open connections to a large number of addresses may be indicative of a worm infection. Anomaly detection raises alert for any activity that doesn't look like normal which makes it suitable for detection of zero day attacks. The problem with anomaly detection model is how to define a model for normal behavior and how to handle evolving normal user behavior. The return of high false positive is another disadvantage of the anomaly detection system. This is the result of its inability to change and adapt over time [10].

2.2 Misuse Detection

A misuse detection model attempts to model abnormal behavior, and compares the network traffic against a signature base of known attacks [11] any match of which clearly indicates system abuse. For example, an HTTP request referring to the cmd.exe file may indicate an attack. A misuse detection technique has reduced false alarms compared to anomaly detection.

Misuse and anomaly detection techniques differ from each other in a way that anomaly detection uses the model of the normal data to detect the anomalous activities whereas misuse detection model uses signatures of some well-known attacks and looks for their occurrence in the network data. The advantage of misuse detection over anomaly detection is higher accuracy and lesser false alarms for the known attacks. The problems with misuse detection models is how to represent the signatures of all possible attacks and how to write signatures that are very different from the normal data pattern. Other problem implicit to the misuse detection model is how to update the signature base when newer attacks appear on the scene.

2.3 Hybrid Approach

Usually signature and anomaly detection are employed together so that they complement each other. This fusion of signature and anomaly detection techniques leads to hybrid approach. This hybrid approach has the combined positives of both the techniques. Survey shows that hybrid technique work better than either of the two techniques. The problem with hybrid approach is the added complexity to lay down the two approaches together to form a complex system, the order in which the two should process the data.

3. MACHINE LEARNING

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. It explores the construction and study of algorithms that can learn from, and make predictions on data [12]. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions. Intrusion detection model is a multinomial classifier problem that can classify network events as normal or attack events, such as Denial of Service (DOS), Probe, U2R, and R2L.

The three prerequisites for Machine Learning are

- Data should be present.
- There should be some pattern in data.
- No simple mathematical model for data.

Machine learning techniques are broadly classified as supervised or unsupervised depending on the presence and absence of the labeled data, and what actually we are trying to predict from the Dataset.

Fig. 1 given below is the pictorial representation of the possible approaches that have been taken to design IDS in last two and an half decade. In the next section we give a brief introduction about each of the machine learning technique.

3.1 Supervised Machine Learning Techniques

Supervised machine learning is the search for algorithms that reason from externally supplied instances to produce general hypothesis, which then make predictions about future instances. In other words, the goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown.

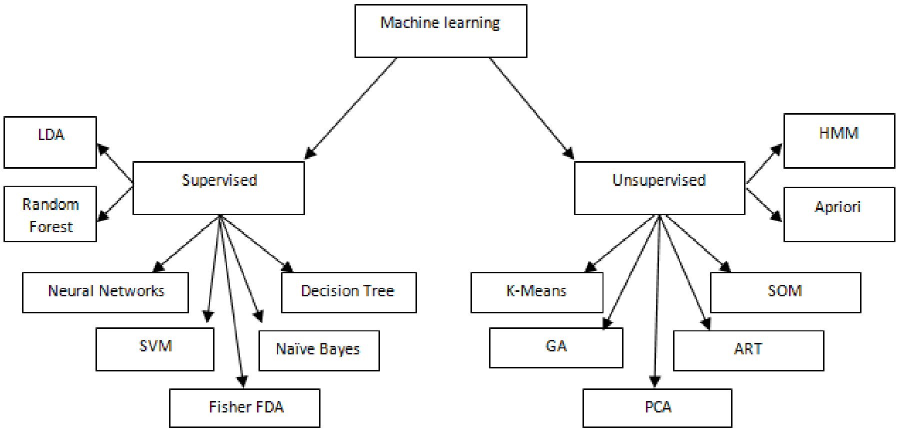

Fig. 1. Machine learning techniques

3.1.1 Decision trees

Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values. The construction of optimal decision tree is a NP hard problem and few heuristic approaches have been put forward. At each level of the decision tree a feature that best divides the tree into subclasses is selected by a variety of ways based on Entropy or Information gain. The division of the tree continues as long as any of the following condition is not met.

- All instances in the training set belong to a single class.
- The maximum tree depth has been reached.
- The best splitting criteria is not greater than a certain threshold.

The selection of the best attribute node is based on the gain ratio *GainRatio*(*S, A*) where S is a set of records and A, *a* non-categorical attribute. This gain defines the expected reduction in entropy due to sorting on A. It is calculated as the following

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (1)$$

In general, if we are given a probability distribution *P* = (*p*$_1$, *p*$_2$ ……*pn*) then the information conveyed by this distribution called the Entropy of *P* is

$$Entropy(P) = -\sum_{i=1}^{n} p_i log_2 p_i \quad (2)$$

If we consider only *Gain*(*S*; *A*) then an attribute with many values will be automatically selected. One solution is to use *GainRatio* instead

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,\ A)} \quad (3)$$

Where

$$SplitInformation(S,A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|} \quad (4)$$

where *Si* is a subset of *S* for which *A* has a value v_i.

3.1.2 Neural networks

Neural Networks is a programming paradigm that has been inspired by the human brain. A neural network is comprised of large number of neurons with each neuron having an input, output and an activation function. The input to a neural network is applied at input layer and in the activation area there some calculations are carried on the input and weights and the output is produced depending weather the sum produced in the activation layer is greater than some predefined threshold. Usually a neural network is laid in the layered approach. The first layer is called the input layer, last layer being called the output layer and other layers are called hidden layers. The optimal number of layers and the number of nodes on each layer is an NP hard problem and are selected by trying different combinations and settling on one that gives the best performance for the problem at hand.

3.1.3 Support vector machines

The latest supervised technique on the scene is Support Vector Machine (SVM). SVM transforms the data in higher dimensions and finds the hyper-plane that best separates the data. A support vector machine is based on the notion of the margin and tries to find the maximum margin between the dataset. SVMs revolve around the notion of a “margin” either side of a hyper-plane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyper-plane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error.

3.1.4 Fuzzy logic

Fuzzy logic is form of knowledge representation suitable for notions that cannot be defined precisely, but which depend upon their contexts. Fuzzy literary means “not clear, distinct, or precise; blurred”, what makes fuzzy logic different from traditional programming approaches is that a fuzzy variable can take any value between zero and one while as a Boolean variable can take either zero or one .Traditional computing logic permits propositions to take a value of truth or falsity while as fuzzy logic allows us to express the degree of truth, which makes it very suitable for modelling real world problems.

3.1.5 Genetic algorithm

The concept of the genetic algorithm comes from the “adaptive survival in natural organisms” [1]. To implement the natural selection and evolution genetic algorithms use the computer system [13]. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions for optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover [14]. Genetic algorithms commence by generating a large population of candidates randomly and in iteration the genetic algorithm replaces the weak solutions with high performing solutions. The performance of a solution is checked against some fitness function, in each iteration the low performing solutions are converted into high performance solutions using mutation and crossover. The solution with low performance is deleted and does not survive to the next iteration.

3.2 Unsupervised Machine Learning

Unsupervised machine learning techniques take on unlabeled dataset and assign the items to certain classes. Absence of the training set and hence cross validation for the cluster analysis marks the difference between clustering and classification. A second important difference is that although most clustering algorithms are phrased in terms of an optimality criterion there is typically no guarantee that the globally optimal solution has been obtained. The reason for this is that typically one must consider all partitions of the data, and for even moderate sample sizes this is not possible, so some heuristic approach is taken. In unsupervised learning we are not concerned about predicting the label for some data item rather our aim is to uncover the hidden groups in data. The discovered groups as such do not have any meaning of their own; and it is left to analyst to derive some meaning from the discovered groups. In cluster analysis there is hardly any hyper-parameter that can be tuned other than the number of clusters the dataset should be divided in.

Once supplied the number of clusters we want to find, the algorithms divide the dataset in the appropriate number of clusters by using some optimization function. Computationally the problem of finding the clusters is difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum [15].

All the clustering algorithms use one or the other distance measure to group the data in certain clusters. The data items grouped together in a cluster are much similar to each other than the data items grouped in different clusters. Given a dataset of n-dimensions clustering approaches calculate some feature and use that feature value to assign the data item to some cluster. Below in Table 1 some of the well-known and commonly used distance measures for the clustering algorithms on two data items x and y consisting of *m-features* is given. Of all the variants given in the table below Euclidean distance is a widely used distance measure.

In the following section some of the clustering techniques used for intrusion detection system are discussed.

3.2.1 *K-means*

One of the mostly used partitioning cluster algorithm is *k-means*. Given M points in N dimensions the *k-means* algorithm divides M points into K clusters so that the within-cluster sum of squares is minimized. It is not practical to require that the solution has minimal sum of squares against all partitions, except when M, N are small and K = 2. We seek instead "local" optima, solutions such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares.

3.2.2 Self-organizing maps

Self-organizing maps (SOMs) were proposed by Kohonen (1995) as a simple method for allowing data to be sorted into groups. The basic idea is to lay out the data on a grid, and to then iteratively move observations (and the centers of the groups) around on that grid, slowly decreasing the amount that centers are moved, and slowly decreasing the number of points considered in the neighborhood of a grid point. A SOM is a sheet-like artificial neural network, whose cells become specifically tuned to various input signal patterns or classes of patterns through an unsupervised learning process. In the basic version, only one cell or local group of cells at a time gives the active response to the current input.

3.2.3 K-mediods

Just like *k-means* algorithm k-Mediods algorithm is also a partitioned and just like *k-means* algorithm k-mediods algorithm also tries to minimize squared error, the distance between points to be in a cluster and a point designated as the center of that cluster. Rather than taking the mean of all the data points as in *k-means* algorithm, k-mediods takes mediods of a finite dataset is a data point from this set, whose average dissimilarity to all the data points is minimal i.e. it is the most centrally located point in the set.

3.2.4 Bayesian clustering

Bayesian Clustering is an unsupervised classification program that uses Bayesian inference to find the most probable classification given the description of cases in the dataset. Although Bayesian Clustering is best suited for the problems where training samples are unlabeled, by ignoring the expert knowledge the system can be used for classifying the labeled data.

3.3 Types of Classifiers

As it is clear from the above two subsections 3.1 & 3.2 that there are various machine learning techniques and they can be laid down in any combination to solve the problem. An approach to solve a problem using a machine learning techniques can be classified as single, ensemble or hybrid depending on the number and the way in which different techniques work to solve a problem.

3.3.1 Single

These are simple most approaches that use a single machine learning technique to solve the problem in hand. This machine learning technique can be any clustering, classification or association techniques. Single learning techniques are easy to grasp fast to implement and easier to implement but do not produce satisfactory results for a problem, hence nowadays are seldom used.

Table 1. Distance measures

Minkowsky	$d(x,y) = \left(\sum_{i=1}^{m} \lvert x_i - y_i \rvert^r\right)^{1/r}$
Manhatan	$d(x,y) = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Chebchev	$d(x,y) = \max_{i=1\ to\ m} \lvert x_i - y_i \rvert$
Euclidean	$d(x,y) = \left(\sum_{i=1}^{m} \lvert x_i - y_i \rvert^2\right)^{1/2}$
Camberra	$d(x,y) = \left(\sum_{i=1}^{m} \left\lvert \frac{x_i - y_i}{x_i + y_i} \right\rvert\right)$
Kendals Rank Correlation	$d(x,y) = \sum_{i=j}^{m} \sum_{j=1}^{i=1} \mathrm{sign}(x_i - x_j)\ \mathrm{sign}(y_i - y_j)$

3.3.2 Ensemble

Another way to solve a problem using a machine learning techniques is to use more than on weak classifiers and then fuse their produced results. Fusion of more than one learning technique together yields better predictive performance than obtained from any of the constituent learning algorithms. Ensemble models achieve performance by combining the opinions of multiple learners. In doing so we often get away using simple classifiers and still achieve great performance. Being inherently parallel in nature ensemble methods can have efficient training and testing time provided we have access to multiple processors. Ensemble methods can be realized in two ways one is training multiple classifiers on the same dataset and the other is training a single classifier on multiple datasets. Once the ensemble is trained then the data item at testing time is assigned to the class which majority of the classifiers point to.

3.3.3 Hybrid

Hybrid approaches combine two machine learning techniques to solve the problem; here the set of machine learning algorithms work in combination rather competing with each other as is the case with ensemble techniques. Hybrid techniques can be laid by cascading two techniques, clustering followed by classification or integration of two different techniques. Clubbing two or more techniques together has the improved performance than there other two counter parts.

4. FEATURE SELECTION TECHNIQUES

Intrusion detection is a classification problem and is based on building the model for that depicts the normal or the anomalous behavior. The data set available for Intrusion detection has large number of features using all the features for the classification is not computationally feasible and may result in reduced performance. So researchers over the years have been devising and using a large number of feature selection algorithms. To note a few Ant Colony Optimization, Cuttlefish Algorithm, Genetic Algorithms are being widely used nowadays. A feature selection algorithm can be classified as filter, wrapper or hybrid method [16].

- **Filter:** Filter methods select the features from the dataset irrespective of the classifier that would be used to build the model for the data. Filter methods take the data with large number of features and select only the best features from this dataset based on some characteristic. These methods use intrinsic characteristics of the dataset to select feature subsets by typically ranking individuals without taking into consideration any data mining algorithms. Filter method analyzes sole features independent of the classifier and decides which features should be kept [17].
- **Wrapper:** Wrapper methods implement a predetermined mining algorithm for evaluating generated subsets of features from the data set. These methods usually have superior performance as they identify features that are better suited to the predetermined mining algorithm. Wrapper based approaches are considered to generate better features, but run much slower and need more computing resources [18].
- **Hybrid Methods:** Some of the researchers have gone a step ahead by incorporating the two feature selection algorithms together such a system is called hybrid system. The hybrid system selects the reliable features for each class but is computationally expensive than either of the two techniques and hard to realize than each of the two techniques.

An alternative to the feature selection is the technique of feature extraction. This technique takes n dimensional data set and transforms it to other dataset which are not the actual features of the original dataset. Approaches [19-21] Take n-dimensional dataset and convert it into one dimension distance vector, then afterwards this distance vector is used for both training and testing purpose. The transformed features are linear combinations of the original attributes.

5. DATASET AND TOOLS AVAILABLE

In the next two subsections the tools and the various versions dataset used for the intrusion detection are discussed briefly. Also the protocol wise attacks present in the dataset are given and in the tabular form what each attribute of the dataset is and type of value it takes is also mentioned.

5.1 Dataset

To check the effectiveness of the techniques a lot of the datasets have been used in practice, most of the works in intrusion detection system have treated the intrusion detection by following a passive approach and once in a while an IDS is fed with the network data on which the IDS applies some mining techniques and uncovers if there are any intrusions. For testing purpose a number of datasets are available for public. Given below a brief introduction about the datasets is given.

5.1.1 KDCup99 dataset

The publicly available and mostly used dataset for intrusion detection is KDCUP99 Data set. This data set is divided into two subsets; training set consisting of 5 million data records and testing set consists of 3 million records. Given in the tables below is the exact count of each type of attack present in KDDcup99 dataset. Each record of this dataset data set has 41 features derived for each connection and a label which classifies connection record as either normal or specific attack type. The features of dataset fall in four categories: ***intrinsic*** features e.g. duration of the connection, type of the protocol (tcp, udp, etc), network service (http, telnet, etc.), etc. The ***content*** feature e.g. number of failed login attempts etc. The ***same host*** features examine established connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to the protocol behavior, service, etc. The *similar same service* features examine the connections in the past two seconds that have the same service as the current connection.

5.1.2 Corrected KDDCup99 dataset

The KDDCup99 dataset is highly redundant records this causes the learning algorithm to be biased to frequent records, and thus prevent them from learning infrequent records which are usually more harmful to networks such as U2R and R2L attacks [22]. In Corrected KddCupp99 dataset all the redundant records have been removed this way the chances of classifier being biased are reduced.

5.1.3 10% KDDCup99 dataset

A complete dataset is seldom used for the training or testing purpose. Rather 10% of the complete dataset is used this dataset has reduced instances of the attacks. Training the classifier on reduced dataset makes if feasible computationally. Below in the Table 2 [22] the count of instances in each of the variants of dataset and the number of particular attacks present in each of the variant is given.

In all the three versions of the dataset the attacks fall in one of the four categories. In Table 3 given below the attack groups and attacks present in KDCUP99 Dataset are listed.

A complete description of each of the 41 features and about the data they take is given Table 4. Features may be continuous or nominal marked by C and N respectively.

As already mentioned above that the KDDCup99 dataset has records of 41 attributes. To provide a clear explanation about how the dataset looks like in Table 5 given below we have listed two records from the dataset one being normal and one being a smurf attack. As can be pointed out from the table some of the features of the dataset are nominal, while some are continuous, the last feature of the record represents the class to which the record belongs to.

5.2 Protocol Wise Analysis of Attacks

In KDDCup99 dataset the simulated attacks can have any of three protocols TCP, UDP, ICMP. Mohammad Khubeb in [23] has done a detailed analysis on the 10% KDDCup99 dataset and

Table 2. Attacks distribution on dataset

Dataset	DoS	U2R	R2L	Probe	Normal	Total
10% KDD	391458	4107	52	1126	97277	494020
Corrected KDD	229853	4166	70	16347	60593	311029
Whole KDD	3883370	41102	52	1126	972780	4898430

Table 3. Attacks types of dataset

Category	Attack types
Probe	nmap, mscan, ipsweep, portsweep,satan, saint
DoS	Back, apache, mailbomb, land, neptune, pod, teardrop, smurf, teardrop, udpstorm
U2R	Perl, rootkit, ps, buffer_overflow, loadmodule, xterm, attack
R2L	Guess_password, imap, ftp_write, imap, multihp, named, phf, snmpgetattack, warezmaster, worm, xsnoop, httptunnel, snmp_guess

Table 4. Complete description of dataset

S. no	Feature name	Data	Description
1	Duration	C	Length of the connection
2	protocol_type	N	Connection protocol
3	service	N	Destination service
4	flag	N	Status flag of the connection
5	src_bytes	C	Bytes sent from source to destination
6	dst_bytes	C	Bytes sent from destination to source
7	land	N	1 if is from/to the same host/port; 0 otherwise
8	wrong_fragment	C	# wrong fragment
9	urgent	C	# urgent packets
10	hot	C	# hot indicators
11	num_failed_logins	C	# failed login in attempts
12	logged_in	N	1 if successfully logged in; 0 otherwise
13	num_compromised	C	# compromised conditions
14	root_shell	N	1 if root shell is obtained; 0 otherwise
15	su_attempted	N	1 if ‘‘su root’’ command attempted; 0 otherwise
16	num_root	C	# root accesses
17	num_file_creations	C	# file creation operations
18	num_shells	C	# shell prompts
19	num_access_files	C	# operations on access control files
20	num_outbound_cmds	C	# outbound commands in an ftp session
21	is_hot_login	N	1 if the login belongs to the hot list; 0 otherwise
22	is_guest_login	N	1 if the login is a guest login; 0 otherwise
23	count	C	# connections to the same host as the current
24	srv_count	C	% connections to the same service as the current connection
25	serror_rate	C	% of connections that have ‘‘SYN’’ errors
26	srv_serror_rate	C	% of connections that have ‘‘SYN’’ errors
27	rerror_rate	C	% of connections that have ‘‘REJ’’
28	srv_rerror_rate	C	% of connections that have ‘‘REJ’’
29	same_srv_rate	C	% of connections to the same service
30	diff_srv_rate	C	% of connections to different services
31	srv_diff_host_rate	C	% of connections to different hosts
32	dst_host_count	C	% count of connections having the same destination host
33	dst_host_srv_count	C	% count of connections having the same destination host and using the same service
34	dst_host_same_srv_rate	C	% of connections having the same destination host and using the same service
35	dst_host_diff_srv_rate	C	% of different services on the current host
36	dst_host_same_src_port_rate	C	% of connections to the current host having the same port
37	dst_host_srv_diff_host_rate	C	% of connections to the same service coming from different hosts
38	dst_host_serror_rate	C	% of connections to the current host that have an SO error
39	dst_host_srv_serror_rate	C	% of connections to the current host and specified service that have an SO error
40	dst_host_rerror_rate	C	% of connections to the current host that have an RST error
41	dst_host_srv_rerror_rate	C	% of connections to the current host and specified service that have an RST error

**C: Continuous, *N: Nominal*

have pointed out that the TCP protocol is most susceptible to the attacks. Table 6 given below list out all the possible attacks for each of the protocol.

10% KddCup99 Dataset Consists of 494020 data instances of these instances 97277 approximately 19.69% of the records are normal record rest of the records depict an attack. An attack can be identified as belonging to any of the four groups DoS, U2R, R2L or Probe. There are 22 different attacks in the training set. Table 7 given below gives the frequency of each attack in the dataset.

5.3 Tools Available

5.3.1 Weka: Data mining software in Java

Weka is a collection of Machine Learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or can be called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new Machine Learning schemes [24]. Weka is preferred by users as it is a free ware and has an easy to use GUI. To use Weka a user need not be an expert in using computer system.

5.3.2 Matlab

Matlab having more than 1 million users across the academia and industry is multi-paradigm numerical computing environment and fourth-generation programming language. The MATLAB application is built around the MATLAB scripting language. [25] Even though Matlab is very rich in features and has very diverse scope than **Weka** at the same time it is difficult to operate and needs a user to have proper understanding of the computer programming.

6. PERFORMANCE PARAMETERS

To check the effectiveness of IDS and document the results a lot of performance metrics have been used below mentioned performance metrics. Researchers have used these metrics to compare their results with already existing approaches [26].

True positive (tp): A positive instance correctly classified as belonging to positive class.

False positive (fp): A negative instance incorrectly classified as belonging to positive class.

True negative (tn): A negative instance correctly classified as a negative example.

False negative (fn): A positive instance incorrectly classified as a negative example.

Table 5. Normal data and smurf attack

Normal	0,tcp,http,SF,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,29,29,1.00,0.00,0.03,0.00,0.00,0.00,0.00,0.00,normal.
Smurf	0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

Table 6. Attacks grouped by protocols

Category	Attack types
UDP	normal, teardrop ,satan, nmap, rootkit
TCP	Normal, neptune, guess_password, land, portsweep, buffer_overflow, phf, warezmaster, ipsweep, multihop, warezclient, perl, back, ftp_write, load_module, satan, spy, imap, rootkit
ICMP	Normal, portsweep, ipsweep, smurf, satan, pod, nmap

Table 7. Frequency of attacks in the dataset

Attack	Count	Attack	Count
Back	2203	Smurf	280790
TearDrop	979	Pod	264
LoadModule	9	Perl	3
Neptune	107201	Warezclient	1020
Rootkit	10	Nmap	231
Phf	4	Imap	12
Satan	1589	Warezmaster	20
Buffer_overflow	30	Portsweep	1040
Ftp_write	8	Guess_password	53
Land	21	Spy	2
Ipsweep	1247	Normal	97277
Multihop	7		

Table 8. Confusion matrix

Actual Class	Predicted class	
	Yes	No
Yes	tp	fn
No	fp	tn

a. **Accuracy (Acc):** In general, the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated.

$$Acc = \frac{tp + tn}{tp + fp + tn + fn}$$

b. **Error Rate(Err):** Misclassification error measures the ratio of incorrect predictions over the total number of instances evaluated

$$Err = \frac{fp + fn}{tp + fp + tn + fn}$$

c. **Sensitivity (sn):** also called the **true positive rate** is the fraction of positive patterns that are correctly classified.

$$sn = \frac{tp}{tp + fn}$$

d. **Specificity (sp):** also called the **true negative rate** or the **precision** in some fields is the fraction of negative patterns that are correctly classified.

$$sp = \frac{tn}{tn + fp}$$

e. **Precision (p):** the positive patterns that are correctly predicted from the total predicted patterns in a positive class

$$p = \frac{tp}{tp + fp}$$

f. **Recall (r):** the fraction of positive patterns that are correctly classified

$$r = \frac{tp}{tp + tn}$$

g. **F-Measure (FM):** This metric represents the harmonic mean between recall and precision values

$$FM = \frac{2 * p * r}{p + r}$$

h. **Geometric-mean (GM):** This metric is used to maximize the *tp* rate and *tn* rate, and simultaneously keeping both rates relatively balanced.

$$GM = \sqrt{tp * tn}$$

i. **Area under Curve (AUC):** AUC value reflects the overall ranking performance of a classifier. For two-class problem given *Sp* the sum of the all positive examples ranked, *np* and *nn* denote the number of positive and negative examples respectively, the AUC value can be calculated as below

$$AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n}$$

j. **Mathews Correlation Coefficient (MCC):** MCC is used to measure the quality of binary classifications. It takes into account true positives, true negatives, false positives and false negatives. MCC can be calculated as below

$$MCC = \frac{(tp * tn) - (fp * fn)}{\sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)}}$$

7. EMERGING PROBLEMS AND PROBABLE SOLUTIONS

Even though a lot of research effort has been spared for intrusion detection using machine learning techniques still there are many problems present and need to be solved to move forward.

7.1 Problems

The existing systems suffer the following problems

- The problem with most of the techniques surveyed is that each of the technique generates too many false alarms. The reason for this could be that the models assume that the user behavior is perfectly observable, legitimate behavior is different from the intrusive behavior and the user usage pattern is steady throughout [10,27,28].
- The low detection rate for the U2R and R2L attacks is another problem present in the currently existing technologies. This could be because the U2R and R2L attacks are very similar to the normal data and are many times misclassified as either the normal data or some other class.

Another reason for low detection rate could be the low frequent occurrence of these classes of attacks which causes the classifier biased to them and hence has reduced detection rate [29-31].

- An IDS by itself is a resource and is prone to be attacked. IDS can be attacked by the attackers and if the attempt to break in the IDS succeeds the network will be left open to the attacker and hence IDS can prove to be single point of failure. There is a long list of the possible attacks on the IDS [7,32].
- An IDS is usually trained on some benchmark dataset and if any implemented in the real environment is all together left in alien conditions. Training and testing the IDS in two different environments reduces the performance.
- There are lots of studies each of them documenting altogether different results even using the same classifier. There is no documentation about the maximum accuracy or the detection rate that an algorithm could attain on a given problem.
- There is no study about which classifier is best for any environment, as there are lot of machine learning techniques available and research has not been up to the point where would compare the classifiers and say that a particular algorithm outnumber some algorithm is all aspects.
- Normal data is very common and anomalous data is rare causing classifier to be biased towards less frequently occurring data items and in case of attacks also some attacks occur very frequently and some occur with less frequency. This restricts us to have an unbiased classifier.
- An IDS raises too many alarms administrator has to deal with, alarming a network administrator for each of the attack independently would lead to too many alarms a manager can deal with. A way to group alarms in groups and raising alarm for each group can be looked into.
- All the techniques surveyed need too much training and testing time which is undesirable and hence none of the techniques is feasible to be implemented in the real environment.
- Although more advanced and sophisticated detection approaches have been developed, very few have focused on feature representation for normal connections and attacks.
- Even though lot of research effort has been put for anomaly detection but still all the present IDS are based on the Misuse detection only because the models devised all are based on the labeled data the thing which we don’t have in then real networks.

7.2 Possible Solutions

Various possible solutions to the problems discussed in 7.1 are

- To minimize the false alarms an IDS should capable of online learning, handling concept drift and should have the ability to be customized to suit any environment.
- To improve the detection rate for R2L and U2R attacks a proper mix of feature extraction, feature selection, data transformation, clustering, classification techniques and the selection of such attributes from the data which are very specific to these two classes of attacks should be taken into consideration.
- To add reliability, scalability and to eliminate single point of failure the feasibility of implementing an IDS as a distributed system can be checked.
- To reduce the biasness of classifier, an IDS should be capable of handling skewed class distribution.
- Reduce the number of alarms an administrator has to deal with by grouping the alarms and issuing a threat for each attack rather for each packet.
- To reduce training and testing time don’t use the features of data as such rather transform the data and represent it as a single point in space and use the transformed single dimensional dataset for training and testing which we believe will work faster than their counterparts.

8. COMPARISON OF WORKS

In this section the surveyed works are put in the tabular form and for each work the model employed, dataset used, features selected, implementation environment and performance metrics are given. A dash (-) in any cell indicates that the author (s) has not mentioned about the feature in their paper. Surveyed works are classified into three tables single given in Table 9, ensemble given in Table 10 and hybrid given in Table 11. At the end of this section an abbreviations of the terms used in the tables is given.

Table 9. Single classifiers

Sl. no	Work	Model employed	Dataset	Selected features/feature selection Algo	Implementing environment	Detection approach	Performance metrics
1.	[33]	DT,NN	10% KddCup99 Dataset	PCA	-	AD	Acc
2.	[34]	PCC	5000 randomly selected normal for training set for testing 92,279 normal & 39,674 attacks elected from KDD'99 data.	-	-	AD	P, R, ROC
3.	[35]	GA	The cmd history of 56 users whose cmd history over one month is more than 500kb	-	-	AD	Acc, Far
4.	[36]	Fuzzy Rough C-Means	1101 Randomly selected data points from KddCup Dataset	-	-	AD	Acc, DR, Far, C
5.	[37]	Ripper	DARPA98	Weighed Dimensions	-	AD,MD	DR
6.	[38]	K-nn	BSM System calls from DARPA98	-	-	AD, MD	DR, ROC
7.	[39]	Statistical Analysis &improved SVM	Real Network Data from Nanjing University of Aeronautics and Astronautics with added attacks.	-	-	AD	ROC
8.	[40]	SOABM	5092 & 6890 random record from KDDCup99 for training and testing randomly	Importance of attribute is marked by the contribution of input made to the construction of DT	-	AD	Acc
9.	[41]	Melhanobis distance Payload Based Anomaly Detector TCP data only	Real Data from Columbia university and DARPA98	-	-	AD	Acc, Far
10.	[42]	PCA	DARPA98 7,000 normal for training and 10000 normal and 396,744attacks for testing	PCA, only 34 numeric features of dataset were used	-	AD, MD	DR, FPR, ROC
11.	[43]	PCA	System call data from University of New Mexico and the Unix command data from AT&T Research lab.	PCA	-	-	Acc, Far, DR
12.	[44]	DT	10% KDDCup99	Cuttlefish optimization algorithm	C# on a Dual Core Machine with 2 GB AM	AD	DR, FP, Acc, ROC
13.	[45]	GMDH	KDDCup99	Feature ranking by	Weka	AD	Acc, R, P, FP, FN, ROC,DR

Sl. no	Work	Model employed	Dataset	Selected features/feature selection Algo	Implementing environment	Detection approach	Performance metrics
				Information, Gain, Gain Ratio			
14.	[46]	GA	KDDCup99	30,000 instances for testing and test sets of 10,000	Disciplus©3 Pentium© IV processor	AD	DR
15.	[47]	GHSOM	KDDCup99, DARPA98	Novel, Muti objective Algo for feature selection	-	AD,MD	DR, ROC
16.	[48]	SOM	10%KDDCup99	3 Feature sets with records from dataset	-	AD	DR, FP
17.	[49]	SOM	10% KDDCup99	Checked 6 basic features	SOM Toolbox and SOM-PAC	AD	FP, DR
18.	[50]	ANN	Probes Attacks from KDDCup99	-	JOONE	MD	DR, FP, FN
19.	[51]	ANN	KDDCup99	IG	JDK	AD	Acc, P, R, Fsore
20.	[52]	SVM	KDDCup99	GFR	LibSVM	AD	Acc, MCC_{avg}
21.	[53]	NB	62986,125973 randomly selected records for training & testing	CFS, IG, GR	WEKA 3.6	MD	Acc, FPR,RMSE, TPR
22.	[54]	GA	8068, 48096 randomly selected for training &testing	-	-	AD, MD	Acc, DR

Table 10. Ensemble classifier

Sno	Work	Model employed	Dataset	Selected features/feature selection	Implementing environment	Detection Approach	Performance metrics
1.	[55]	3 NN having 5 outputs 30 inputs and one hidden layer of 5 nodes	725 ftp connections present in KDDCup99 for training and 7436 for testing	11 features having constant value for every Ftp connection have been eliminated	-	AD	Overall Classification Error, Classification Cost
2.	[56]	HMM, statistical method and rule base Method	13 megabytes of BSM audit data and 840 kb of PACCT audit data have been collected from 16,470commands Attacks are Buffer overflow and Dos	-	-	AD	DR, FP, ERR, ROC
3.	[57]	DT, NB, rule learner, SVM and	live lpr, live lpr MIT, synthetic sendmail, synthetic sendmail CERT, and “denial of	-	-	MD & AD	Acc, DR,FPR

Sno	Work	Model employed	Dataset	Selected features/feature selection	Implementing environment	Detection Approach	Performance metrics
		NB & K-means Clustering	service attack” of UNM & MIT LL system call sequences				
4.	[58]	LGP	11982randomly generated points from kddcup99 5092, 6890 for training and testing respt.	Feature Ranking, have found out the important features for each class of attack	C++	MD , AD	Acc
5.	[59]	GA+FL	1000 randomly selected samples for training from kddcup99 and 10000 randomly selected samples for testing	-	-	MD	CR, DR, FR
6.	[60]	SVM, ANN, MARS	11982 randomly generated points from kddcup99 5092, 6890 for training and testing respectively.	-	-	AD	Acc

Table 11. Hybrid classifiers

Sl. no	Work	Model employed	Dataset	Selected features/feature selection	Implementing environment	Detection approach	Performance metrics
1.	[61]	SVM, ACN	A subset of KDDCup99 Dataset	-	-	AD	DR, FP, FN
2.	[62]	DT,SVM	Modified KDDCup99 Dataset	-	Weka 3.6 & LibSVM	AD,MD	DR, ROC
3.	[63]	SA + DT,	KDDCup99	SA+ SVM	-	AD	DR
4.	[64]	DT,ANN, Ripper Rule	KDDCup99/ Reliability Lab Data	Information Gain 12 Feature Extracted from packet header	-	MD	DR
5.	[65]	FC + ANN	Randomly selected 18,285 from KDDCup99	-	Matlab 2007b	AD	P, R, F-measure
6.	[66]	K-means + SVM	KDDCup99	-	-	AD	DR
7.	[67]	DGSOT + SVM	DARPA98	-	-	AD	FP, FN, Acc
8.	[68]	TCM-KNN	KDDCup99	Chi-Square method and SVM attribute evaluation method	Weka 3.5	AD	TP, FP
9.	[69]	SOM + PCA-ANN	DARPA98	PCA	-	AD, MD	DR, Far, FP
10.	[70]	C means Clustering + ANN, RBF	KDDCup99	-	-	AD, MD	DR, FP
11.	[71]	FL + AR	10% Corrected KDDCup99	-	Standard C	AD	DR, FP
12.	[72]	BC + DT	10000 randomly selected normal data and all u2r and r2l ofKDDCup99	InfoGain	Weka 3.4, AutoClass	AD	DR, RT

Sl. no	Work	Model employed	Dataset	Selected features/feature selection	Implementing environment	Detection approach	Performance metrics
13.	[73]	GA + FL	Randomly selected 2% records of 10% KDDCup99	-	GAlib, Oracle database 8i, Visual Basic	AD, MD	DR
14.	[74]	GA + SVM, SOFM	DARPA98, Live Dataset 100,000 normal data packets & 1000–1500 attacks	GA	LibSVM	AD	DR, FP, FN
15.	[75]	SVM + ANN	DARPA98	-	Matlab 6.1 NN Toolbox software	AD	DR, FP
16.	[76]	DT + SVM	KDDCup99	-	-	AD + MD	Acc
17.	[77]	ANN + Fuzzy Inference	DARPA98	DT	-	MD	Acc
18.	[78]	Fuzzy System, rule based Expert System	KDDCup99	-	FuzzyCLIPS	AD + MD	FPR
19.	[79]	FNT	11,982 randomly generated records DARPA98	Important features are selected for each class of attacks	-	AD	DR, FPR, FNR, Acc
20.	[80]	SOM + DT	KDD Cup 99	6 basic features from the dataset		AD + MD	DR,FPR, MR
21.	[81]	Geometric Technique, k-nn, SVM	DARPA98, KDDCUP99	-	-	AD	DR, FPR, ROC
22.	[82]	FL	DARPA98	A program that extracts and combines 11 features	-	AD	Acc
23.	[83]	Unsupervised Clustering Technique	DARPA98 & KDDCUP99	-	-	AD	DR, FAR
24.	[84]	BPN, RBF, k-NN and SVM	Training data from information Systems Technology Group (IST) of MIT Lincoln Lab	GA	Matlab, SVM light Package	AD	CR, FP, FN
25.	[85]	NF	10% KDDCup99	-	-	AD	DR,FAR, Acc
26.	[86]	Fuzzy rule-based system, GA	10% KDDCup99	Info Gain, Gain Ratio, Chi-square, Relief-F	-	AD	R, P, Acc, F-measure, ROC
27.	[87]	Rough Set Classification	KDDCup99 Randomly selected various attack groups attacks of each group	GA	MS Visual C++ 6.0 language is C and C++, LIBSVM	AD + MD	DR, MCR, TT

Sl. no	Work	Model employed	Dataset	Selected features/feature selection	Implementing environment	Detection approach	Performance metrics
28.	[88]	MLP	20,055 randomly selected normal and attacks connections from DARPA98 only two attacks SYN Flood (Neptune) and Satan	19 features describing properties of connections to the same host in last two seconds	MATLAB Neural Network Toolbox	AD	CR
29.	[89]	DT, NN	10% KDDCup99	GA	Java	AD	TPR, FPR, P, R,F-measure
30.	[90]	NBTREE	10%KDDCup99	FCBR	Java, Weka, SQL	MD	Acc, ERR
31.	[91]	Decision Table+PART	KDDCup99	CFS, GR,IG	Weka	MD	Acc, RMSE, TPR, FPR, F-M-measure
32.	[92]	CCA-S	A random subset of DARPA98	-	AnswerTree from SPSS	MD	FAR, ROC
33.	[93]	RBF NN + SVM	25192 records from KDDCup99	PCA	JAVA	AD, MD	Dr, FPR, F-Value, P, R, RMSE
34.	[94]	BIRCH + SVM	Whole KDDCup99	Leave one out	LibSVM, JAVA	AD	Acc, FPR, TT
35.	[95]	HNB	10%KDDCup99	CFS, CBF, INT	-	AD, MD	Acc, Err
36.	[96]	TAN + REP	10%KDDCup99	-	-	AD	TPR, Acc
37.	[97]	K-means + DT	15000 & 2500 KDDCup99 records for training and testing respt.	-	Weka 3.5	AD	Acc, FPR, F-Measure, TPR
38.	[98]	PSO + ARTMAP	10%KDDCup99 & corrected Dataset for training and testing	FARM	MATLAB R2010	MD	DR, FAR, CR
39.	[99]	FL + GNP	KDDCup99 & DAPRA98	-	-	AD, MD	DR
40.	[100]	SSO	10%KDDCup99	IDS-RS	-	AD, MD	CAR

1. Acc: Accuracy, 2. AD: Anomaly Detection, 3. ACN: Ant Colony Network, 4. AR: Association Rule, 5. BPN: Back Propagation Network, 6. CBF: Consistency Based Filter, 7. CCA-S: Clustering and Classification Algorithm Supervised, 8. CFS: Correlation Based Feature Selection, 9. CR: Classification Rate, 10. DGSOT: Dynamically Growing Self-Organizing Tree, 11. DR: Detection Rate, 12 DT: Decision Tree, 13. ERR: Error Rate, 14. Far: False Alarm Rate, 15. FC: Fuzzy Clustering, 16. FCBF: Fast Correlation Based Filter, 17. FL: Fuzzy Logic, 18. FN: False Negative, 19. FNT: Flexible Neural Tree, 20. FP: False Positive, 21. FPR: False Positive Rate, 22. GA: Genetic Algorithm, 23. GFR: Gradual Feature Removal, 24. GHSOM Growing Hierarchical Self-Organizing Maps, 25. GMDH: Group Method for Data Handling, 26. HNB: Hidden Naïve Bayes, 27. IDS-RS: Intelligent dynamic swarm based Rough Set, 28. IG: Information Gain, 29. INT: INTERACT, 30. LGP: Linear Genetic Programming, 31. MARS:, 32. MCR: Miss Classification Rate, 33. MD: Misuse Detection, 34. MLP: Multi Layer Perceptron, 35. NB: Naïve bayes. 36. NBTREE: Naïve Bayes and Decision tree, 37. NF: Neuro Fuzzy, 38. NN: Neural Networks, 39. PCA: Principal Component Analysis, 40 PCC Principal Component Classifier, 41 K-nn: K nearest neighbor, 42. R: Recall, 43. RBF: Radial Bayes Function, 44. REP: Reduced Error Pruning, 45. ROC: Receiver Operating Curve, 46. SA: Simulated Annealing, 47. SOABM: Self Organized Ant based, 48.Clustering Method, 49. SOM: Self Organizing Map, 50. SOFM: Self-Organized Feature Map, 51 SPegasos, 52. SSO: Simplified Swarm Optimization, 53. SVM: Support Vector Machine, 54. TAN: Tree Augmented Naïve Bayes, 55. TCM-KNN: Transductive Confidence Machines for K-Nearest Neighbors, 56. TP: True Positive, 57. TN: True Negative

9. CONCLUSION

In this work a survey of intrusion detection systems using machine learning techniques was given. A lots of relates works were surveyed and classified into three groups single, ensemble or hybrid and for each work the dataset used, environment in which implemented, feature selection if any and the performance measures checked were documented in tabular form. A complete list of the attacks present in the KDDCup99 dataset is also given. The thorough survey of the works has revealed that the hybrid machine learning techniques with the proper feature selection algorithm out class there single or ensemble counterparts. Also in this paper an effort was made to point out problems pertaining to the current system and directions for future research were also provided.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. CF Tsai, YF Hsu, CY Lin, WY Lin. Intrusion detection by machine learning: A review. 10, s.l. : Elsevier, Expert Systems with Applications. 2009;36:11994-12000.
2. Graham Robert. FAQ: Network intrusion detection systems. 2000;8:3.
3. Heady Richard, et al. The architecture of a network level intrusion detection system. Department of Computer Science, College of Engineering: University of New Mexico; 1990.
4. Venter HS, Jan HP Eloff. A taxonomy for information security technologies. 4, s.l.: Elsevier, Computers & Security. 2003;22: 299-307.
5. Sangkatsanee Phurivit, Naruemon Wattanapongsakorn, Chalermpol Charnsripinyo. Practical real-time intrusion detection using machine learning approaches. 18, s.l.: Elsevier, Computer Communications. 2011;34:2227-2235.
6. Chen ChiaMei, YaLin Chen, HsiaoChung Lin. An efficient network intrusion detection. 4, s.l.: Elsevier, Computer communications. 2010;33:477-484.
7. Ptacek Thomas H, Timothy N Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. s.l. : Secure Networks Inc Calgary Alberta; 1998.
8. Stallings William. Network and internetwork security: Principles and practice. Englewood Cliffs: Prentice Hall. 1995;1.
9. Verwoerd, Theuns, Ray Hunt. Intrusion detection techniques and approaches. 15, s.l.: Elsevier, Computer Communications. 2002;25:1356-1365.
10. Shun Julian, Heidar Malki. Network intrusion detection system using neural networks. s.l.: IEEE, ICNC'08. Fourth International Conference. 2008;5.
11. Anonymous. Intrusion detection FAQ. May 19; 2010.
    Available:http://www.sans.org/
    Available:http://www.sans.org/security-resources/idfaq/
    [Cited: august 08, 2015.]
12. Machine learning. [Online] August 6; 2015.
    Available:https://en.wikipedia.org/wiki/Machine_learning
    [Cited: August 8, 2015.]
13. Koza, John R. Genetic programming: On the programming of computers by means of natural selection. s.l.: MIT Press. 1992; 1.
14. Genetic algorithm. [Online] August 07; 2015.
    Available:https://en.wikipedia.org/wiki/Genetic_algorithm
    [Cited: August 10, 2015.]
15. *k-means* clustering. [Online] July 30; 2015.
    Available:https://en.wikipedia.org/wiki/K-means_clustering
    [Cited: July 8, 2015.]
16. Kohavi, Ron, George H John. Wrappers for feature subset selection. 1, s.l.: Elsevier, Artificial Intelligence. 1997;97:273-324.
17. Yinhui LI, et al. An efficient intrusion detection system based on support vector machines and gradually feature removal method. 1, s.l.: Elsevier, Expert Systems with Applications. 2012;39:424-430.
18. Hall Mark A, Lloyd A Smith. [ed.]. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. IEEE. FLAIRS Conference. 1999; 235-239.
19. Luo Bin, Jingbo Xia. A novel intrusion detection system based on feature generation with visualization strategy. 9, s.l.: Elsevier, Expert Systems with Applications. 2014;41:4139-4147.
20. Lin Wei-Chao, Shih-Wen Ke, Chih-Fong Tsai. CANN: An intrusion detection system

based on combining cluster centers and nearest neighbors. Knowledge-Based Systems. 2015;78:13-21.

21. Tsai Chih-Fong, Chia-Ying Lin. A triangle area based nearest neighbors approach to intrusion detection. 1, s.l.: Elsevier, Pattern Recognition. 2010;43:222-229.
22. Jeya P Gifty, Ravichandran M, Ravichandran CS. Efficient classifier for R 2 L and U 2 R attacks. 21, International Journal of Computer Applications. 2012; 45.
23. Siddiqui, Mohammad Khubeb, Shams Naahid. Analysis of KDD CUP 99 dataset using clustering based data mining. International Journal of Database Theory and Application. 2013;6:23-34.
24. Weka (machine learning). [Online] August 07; 2015. Available:https://en.wikipedia.org/wiki/Weka_(machine_learning) [Cited: August 08, 2015.]
25. MATLAB. [Online] August 07; 2015. Available:https://en.wikipedia.org/wiki/MATLAB. [Cited: August 08, 2015.]
26. Liao Yihua, Rao Vemuri V, Alejandro Pasos. Adaptive anomaly detection with evolving connectionist systems. 1, s.l.: Elsevier, Journal of Network and Computer Applications. 2007;30:60-80.
27. Clifton Chris, Gary Gengo. Developing cutsom intrusion detection system filters using data mining. MILCOM : IEEE, 21st Century Military Communications. 2000;1.
28. Kumari, Ranjitha S, Krishna Kumari P. Adaptive Anomaly intrusion detection system using optimized hoeffding tree and online adaboost algorithm. 1, World Applied Sciences Journal. 2015;33:102-108.
29. Lin WeiChao, ShihWen Ke, ChihFong Tsai. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. s.l.: Elsevier, Knowledge-Based Systems. 2015;78: 13-21.
30. Horng Shi-Jinn, et al. A novel intrusion detection system based on hierarchical clustering and support vector machines. s.l.: Elsevier, Expert systems with Applications. 2011;38:306-313.
31. Wang Shun-Sheng, et al. An integrated intrusion detection system for cluster-based wireless sensor networks. 12, s.l.: Elsevier, Expert Systems with Applications. 2011;38:15234-15243.
32. Cheng Tsung-Huan. Evasion techniques: Sneaking through your intrusion detection/prevention system. 4, s.l.: IEEE, Communications Surveys & Tutorials. 2012;14:1011-1020.
33. Bouzida Yacine, et al. Efficient intrusion detection using principal component analysis. France : s.n., 3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe; 2004.
34. Shyu Mei-Ling, et al. A novel anomaly detection scheme based on principal component classifier. s.l. : Miami Univ Coral Gables FL Dept of Electrical and Computer Engineering; 2003.
35. Balajinath B, Raghavan SV. Intrusion detection through learning behavior model. Computer Communications. 12, s.l.: Elsevier, Computer Communications. 2001;24:1202-1212.
36. Chimphlee Witcha, et al. Anomaly-based intrusion detection using fuzzy rough clustering. s.l.: IEEE. Hybrid Information Technology. ICHIT'06. International Conference. 2006;1:329-334.
37. Fan Wei, et al. Using artificial anomalies to detect unknown and known network intrusions. 5, s.l.: Springer, Knowledge and Information Systems. 2004;6:507-527.
38. Liao Yihua, Rao Vemuri V. Use of k-nearest neighbor classifier for intrusion detection. 5, s.l. : Elsevier, Computers & Security. 2002;21:439-448.
39. Tian Ming, et al. Using statistical analysis and support vector machine classification to detect complicated attacks. In Machine Learning and Cybernetics. Proceedings of 2004 International Conference. 2004;5: 2747-2752.
40. Ramos Vitorino, Ajith Abraham. ANTIDS: self organized ant-based Clustering model for intrusion detection system. s.l. : Springer, Soft Computing as Trans-disciplinary Science and Technology. 2005;977-986.
41. Wang Ke, Salvatore J Stolfo. Anomalous payload-based network sion detection. Heidelberg: Springer, Recent Advances in Intrusion Detection. 2004;203-222.
42. Wang Wei, Roberto Battiti. Identifying intrusions in computer networks with principal component analysis. s.l. : IEEE. Availability, Reliability and Security, 2006. ARES. The First International Conference. 2006;8.
43. Wang Wei, Xiaohong Guan, Xiangliang Zhang. A novel intrusion detection method

based on principle component analysis in computer security. Berlin Heidelberg: Springer, Advances in Neural Networks-ISNN. 2004;657-662.

44. Eesa Adel Sabry, Zeynep Orman, Adnan Mohsin Abdulazeez Brifcani. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. 5, s.l. : Elsevier, Expert Systems with Applications. 2015;42:2670-2679.
45. Baig Zubair A, Sadiq M Sait, AbdulRahman Shaheen. GMDH-based networks for intelligent intrusion detection. 7, s.l. : Elsevier, Engineering Applications of Artificial Intelligence. 2013;26:1731-1740.
46. Hansen James V, et al. Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. 4, s.l. : Elsevier, Decision Support Systems. 2007;43:1362-1374.
47. De la Hoz, Emiro, et al. Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. s.l.: Elsevier, Knowledge-Based Systems. 2014;71:322-338.
48. Sarasamma, Suseela T, Qiuming Zhu, Julie Huff. Hierarchical Kohonenen net for anomaly detection in network security. 2, s.l.: IEEE, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions. 2005;35:302-312.
49. Kayacik H Gunes, Nur Zincir-Heywood A, Malcolm I. A hierarchical SOM-based intrusion detection system. Heywood. 4, s.l. : Elsevier, Engineering Applications of Artificial Intelligence. 2007;20:439-451.
50. Iftikhar Ahmadd, Azween B Abdullah, Abdullah S Alghamdi. Application of artificial neural network in detection of probing attacks. Kuala Lumpur, Malaysia: IEEE. IEEE Symposium on Industrial Electronics and applications; 2009.
51. Bhavin Shah, Bushan H Trivedi. Reducing features of KDD CUP 1999 dataset for anomaly detection using back propagation neural network. s.l. : IEEE. Fifth International Conference on Advanced Computing & Communication Technologies; 2015.
52. Yinhui Li, et al. An efficient intrusion detection system based on support vector machines and gradually feature removal method. 1, s.l. : Elsevier, Expert Systems with Applications. 2012;39:424-430.
53. Saurabh Mukherjee, Neelam Sharma. Intrusion detection using naive bayes classifier with feature reduction. s.l. : Elsevier, Procedia Techonology. 2012;4: 119-128.
54. Lu Nannan. An efficient class association rule-pruning method for unified intrusion detection system using genetic algorithm. 2, s.l.: John Wiley & Sons, Inc., IEEJ Transactions on Electrical and Electronic Engineering. 2013;8:164-172.
55. Giacinto Giorgio, Fabio Roli. Intrusion detection in computer networks by multiple classifier systems. s.l. : IEEE. Pattern Recognition. Proceedings. 16th International Conference. 2002;2:390-393.
56. Han Sang-Jun, Sung-Bae Cho. Detecting intrusion with rule-based integration of multiple models. 7, s.l. : Elsevier, Computers & Security. 2003;22.
57. Kang Dae-Ki, Doug Fuller, Vasant Honavar. Learning classifiers for misuse and anomaly detection using a bag of system calls representation. s.l. : IEEE, 2005. Information Assurance Workshop. IAW'05. Proceedings from the Sixth Annual IEEE SMC. 2005;118-125.
58. Mukkamala Srinivas, Andrew H Sung, Ajith Abraham. Modeling intrusion detection systems using linear genetic programming approach. Berlin Heidelberg: Springer, Innovations in Applied Artificial Intelligence. 2004;633-642.
59. Abadeh, Mohammad Saniee, et al. A parallel genetic local search algorithm for intrusion detection in computer networks. 8, s.l. : Elsevier, Engineering Applications of Artificial Intelligence. 2007;20:1058-1069.
60. Mukkamala Srinivas, Andrew H Sung, Ajith Abraham. Intrusion detection using an ensemble of intelligent paradigms. 2 , s.l.: Elsevier, Journal of Network and Computer Applications. 2005;28:167-182.
61. Feng Wenying, et al. Mining network data for intrusion detection through combining SVMs with ant colony networks. s.l.: Elsevier, Future Generation Computer Systems. 2014;37:127-140.
62. Kim Gisung, Seungmin Lee, Sehun Kim. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. 4, s.l. : Elsevier, Expert Systems with Applications. 2014;41:1690-1700.
63. Lin Shih-Wei, et al. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. 10,

s.l.: Elsevier, Applied Soft Computing. 2012;12:3285-3290.

64. Sangkatsanee Phurivit, Naruemon Wattanapongsakorn, Chalermpol Charnsripinyo. Practical real-time intrusion detection using machine learning approaches. 18, s.l.: Elsevier, Computer Communications. 2011;34:2227-2235.

65. Wang Gang, et al. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. 9, s.l.: Elsevier, Expert Systems with Applications. 2010;37:6225-6232.

66. Giacinto Giorgio, et al. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. 1, s.l.: Elsevier, Information Fusion. 2008;9:69-82.

67. Khan Latifur, Mamoun Awad, Bhavani Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. 4, s.l.: ACM, The VLDB Journal—The International Journal on Very Large Data Bases. 2007;16: 507-521.

68. Li Yang, Li Guo. An active learning based TCM-KNN algorithm for supervised network intrusion detection. 7, s.l.: Elsevier, Computers & Security. 2007;26.

69. Liu Guisong, Zhang Yi, Shangming Yang. A hierarchical intrusion detection model based on the PCA neural networks. 7, s.l.: Elsevier, Neurocomputing. 2007;70:1561-1568.

70. Zhang Chunlin, Ju Jiang, Mohamed Kamel. Intrusion detection using hierarchical neural networks. 6, s.l. : Elsevier, Pattern Recognition Letters. 2005;26.

71. Tajbakhsh Arman, Mohammad Rahmati, Abdolreza Mirzaei. Intrusion detection using fuzzy association rules. 2, s.l. : Elsevier, Applied Soft Computing. 2009;9.

72. Xiang Cheng, Png Chin Yong, Lim Swee Meng. Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. 7, s.l.: Elsevier, Pattern Recognition Letters. 2008;29:918-924.

73. Özyer Tansel, Reda Alhajj, Ken Barker. Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening. 1, s.l. : Elsevier, Journal of Network and Computer Applications. 2007;30:99-113.

74. Shon Taeshik, Jongsub Moon. A hybrid machine learning approach to network anomaly detection. 18, s.l.: Elsevier, Information Sciences. 2007;177:3799-3821.

75. Chen WunHwa, ShengHsun Hsu, HwangPin Shen. Application of SVM and ANN for intrusion detection. 10, s.l. : Elsevier, Computers & Operations Research. 2005;32:2617-2634.

76. Peddabachigari, Sandhya, Ajith Abraham, Johnson Thomas. Intrusion detection systems using decision trees and support vector machines. 3, USA: Springer, International Journal of Applied Science and Computations. 2004;11.

77. Chavan Sampada, et al. Adaptive neuro-fuzzy intrusion detection systems. Information Technology: Coding and Computing. s.l. : IEEE. ITCC. 2004;1:70-74.

78. Bridges Susan M, Rayford B Vaughn. Intrusion detection via fuzzy data mining. 12th Annual Canadian Information Technology Security Symposium. 2000; 109-122.

79. Chen Yuehui, Ajith Abraham, Bo Yang. Hybrid flexible neural-tree-based intrusion detection systems. 4, s.l.: Wiley, International Journal of Intelligent Systems. 2007;22:337-352.

80. Depren Ozgur, et al. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. 4, s.l.: Elsevier. 2005;29:713-722.

81. Eskin Eleazarl. A geometric framework for unsupervised anomaly detection. US: Springer, Applications of Data Mining in Computer Security. 2002;77-101.

82. Florez German, Susan M Bridges, Rayford B Vaughn. An improved algorithm for fuzzy data mining for intrusion detection. North American: IEEE. Fuzzy Information Processing Society. 2002;457-462.

83. Jiang Sheng Yi, et al. A clustering-based method for unsupervised intrusion detections. 7, s.l.: Elsevier, Pattern Recognition Letters. 2006;27:802-810.

84. Shon T, Kovah X, Moon J. Applying genetic algorithm for classifying anomalous TCP/IP packets. 16, s.l.: Elsevier, Neurocomputing. 2006;69:2429-2433.

85. Toosi Adel Nadjaran, Mohsen Kahani. A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. 10, s.l.: Elsevier, Computer Communications. 2007;30:2201-2212.

86. Tsang Chi-Ho, Sam Kwong, Hanli Wang. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques

for anomaly intrusion detection. 9, s.l. : Elsevier, Pattern Recognition. 2007;40: 2373-2391.

87. Zhang Lianhua. Intrusion detection using rough set classification. 9, s.l. : Springer, Journal of Zhejiang University Science. 2004;5:1076-1086.
88. Moradi Mehdi, Mohammad Zulkernine. A neural network based system for intrusion detection and classification of attacks. s.l. : IEEE. IEEE international conference on advances in intelligent systems-theory and applications; 2004.
89. Sindhu, Siva S Sivatha, Geetha S, Kannan A. Decision tree based light weight intrusion detection using a wrapper approach. 1, s.l. : Elsevier, Expert Systems with applications. 2012;39:129-141.
90. Datta H Deshmukh, Tushar Ghorpade, Puja Padiya. Improving Classification accuracy using perprocessing and machine learnming algorithms on NSL-KDD Dataset. Mumbai, India: IEEE. International Confrence on Communication, Information & Computing Technology; 2015.
91. Shilpa Bahl, Sudhir Kumar Sharma. Improving classifiaction accuracy of intrusion detection system using feature subset selection. s.l.: IEEE. Fifth International Conference on Advanced Computing & Communication Techonologies; 2015.
92. Li Nong, Yem Xiangyang. A scalable clustering technique for intrusion signature reduction. Newyork : IEEE. Workshop on Information Accuracy and Security, United States Miitary Academy; 2001.
93. Mrutyunjaya Panda, Ajith Abrahim, Mannas Ranjan Patra. A hybrid intelligent approach for network intrusion detection. s.l.: Elsevier. Procedia Engineering. 2012; 1-9.
94. Shi-Jinn Horng, et al. A novel intrusion detection system based on hierarchical clustering and support vector machines. 1, s.l.: Elsevier, Expert Systems with Applications. 2011;38:306-313.
95. Koc Levent, Thomas A Mazzuchi, Shahram Sarkani. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. 18, s.l.: Elsevier, Expert Systems with Applications. 2012;39:13492-13500.
96. Mradul Dhakar, Akhilesh Tiwari. A novel data mining based hybrid intrusion detection framework. 1, s.l. : World Academic Press, Journal of Information and Computing Science. 2014;9.
97. Amuthan Prabakar Muniyandi, Rajeswari R, Rajaram R. Network anomaly detection by cascading K-means clustering and C4.5 decision tree algorithm. s.l.: Elsevier, Prodedia Engineering. 2012;30:174-182.
98. Sheikhan Mansour, Maryam Sharifi Rad. Using particle swarm optimization in fuzzy association rules-based feature selection and fuzzy ARTMAP-based attack recognition. 7, s.l.: John Wiley & Sons, Ltd, Security and Communication Networks. 2013;6:797-811.
99. Mabu Shingo. An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. 1, s.l.: Elsevier, Systems, Man, and Cybernetics, Part C: Applications and Reviews. 2011;41:130-139.
100. Chung Yuk Ying, Noorhaniza Wahid. A hybrid network intrusion detection system using simplified swarm optimization (SSO). 9, s.l.: Elsevier, 3014-3022, Applied Soft Computing. 2012;12.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/13804