ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/325486854

A Hybrid Behavioral-Based Cyber Intrusion Detection System

Article · June 2018

CITATIONS
0

READS
95

2 authors, including:

Henock Mulugeta
Ambo University
7 PUBLICATIONS **9** CITATIONS

SEE PROFILE

All content following this page was uploaded by Henock Mulugeta on 02 June 2018.

The user has requested enhancement of the downloaded file.

A hybrid behavioural-based cyber intrusion detection system

Alemtsehay Adhanom*

Electrical and Computer Engineering Department,
Addis Ababa Institute of Technology,
Addis Ababa University, Ethiopia
Email: alemasm20@gmail.com
*Corresponding author

Henock M. Melaku

Department of Computer Science,
Institute of Technology,
Ambo University,
Ambo, Ethiopia
Email: henock.mulugeta@ambou.edu.et
Email: henockmulugeta26@yahoo.com

Abstract: The experience of deploying intrusion detection system (IDS) for securing computer system is being matured. There are knowledge-based (misuse) and anomaly IDS. In knowledge-based IDS, prior knowledge of the attack is needed for detection and during anomaly, behaviour of normal data is studied, when new data is arrived and there is a deviation, it is considered as an attack. In this thesis, we present a hybrid intrusion detection system called behavioural-based cyber intrusion detection system, based on two data mining algorithms, decision tree and association rule mining. The decision tree algorithm is used to detect misuse intrusions but it considers new attacks as normal. Association rule mining works by using the normal output of decision tree as input for further detection. Further, we implement the proposed model using java programming language. We have used a reduced and enhanced non-redundant NSL_KDD dataset for training and testing. Evaluation results show that it provides improved detection rate and lower false alarm rates.

Keywords: intrusion detection system; IDS; knowledge discovery data mining; genetic algorithm.

Reference to this paper should be made as follows: Adhanom, A. and Melaku, H.M. (xxxx) 'A hybrid behavioural-based cyber intrusion detection system', *Int. J. Communication Networks and Distributed Systems*, Vol. X, No. Y, pp.xxx–xxx.

Biographical notes: Alemtsehay Adhanom received her MSc degree from the Addis Ababa Institute of Technology, School of Electrical and Computer Engineering, AAU. Her research interest is computer and network security, intrusion detection system.

Copyright © 20XX Inderscience Enterprises Ltd.

Henock M. Melaku received his PhD degree from the Addis Ababa Institute of Technology (AAiT), Addis Ababa University, in Computer Engineering, 2014. Currently, he is an Assistant Professor at the Institute of Technology, Ambo University and AAiT. His research interests are wireless networks, mobile and ubiquitous computing, mobile ad hoc networks, wireless sensor networks, TCP, routing protocols, MAC and security issues in mobile and wireless sensor networks.

1 Introduction

Information security technology is an essential component for protecting public and private computing infrastructures. With the widespread utilisation of information technology applications, organisations are becoming more aware of the security threats to their resources. No matter how strict the security policies and mechanisms are, more organisations are becoming easily influenced to a wide range of security breaches against their electronic resources.

Computer security (cyber security) is the process of ensuring confidentiality, integrity and availability of computing systems (Airehrour et al., 2016; Shah and Agrawal, 2016; Ahmed, 2017; Leloglu, 2016). Lack of security results from a failure of one of these three properties. The traditional and static prevention techniques such as user authentication, data encryption, virtual private network (VPN), avoiding programming errors and firewalls are used as the first line of defence for computer security. If a password is weak and is compromised, user authentication cannot prevent unauthorised use; firewalls are vulnerable to errors in configuration and ambiguous or undefined security policies. They are generally unable to protect against malicious mobile code, insider attacks and unsecured modems. Intrusion detection is therefore a dynamic one which is required as an additional wall for protecting systems.

The concept of intrusion detection system (IDS) was proposed and available in various literature (Al-Jarrah et al., 2016; Ahmed, 2017; Sforzin et al., 2016), a study outlining ways to improve computer security auditing and surveillance at customer sites. The original idea behind automated ID is often credited to him for his paper on "How to use accounting audit files to detect unauthorised access". This ID study paved the way as a form of misuse detection for mainframe systems. The first task was to define what threats existed before designing an IDS, it was necessary to understand the types of threats and attacks that could be mounted against computer systems and how to recognise them in an audit data (Alrawais et al., 2107; Ayman et al., 2014; Omran and Panda, 2016; Sultana and Jabbar, 2016). Each malicious activity or attack has a specific pattern. The patterns of only some of the attacks are known whereas the other attacks only show some deviation from the normal patterns. Therefore, the techniques used for detecting intrusions are based on whether the patterns of the attacks are known or unknown. The two main techniques used are: anomaly and misuse detections (Arrington et al., 2016; Omran and Panda, 2016; Sultana and Jabbar, 2016). Anomaly detection technique is while IDS has knowledge of normal behaviour so it searches for anomalous behaviour or

deviations from the established baseline. It's most apparent drawback is high false alarms rates, it does offer detections of unknown intrusions and new exploits; on the other hand, misuse detection technique is while IDS has knowledge of suspicious behaviour and searches activity that violates stated policies. It also means looking for known malicious or unwanted behaviour. In fact, its main features are its efficiency and comparably low false alarm rate.

IDS can be host-based or network-based systems (Bertino et al., 2016; Omran and Panda, 2016; Chiche and Meshesha, 2017). A host-based ID uses data from a single host to detect signs of intrusion as the packets enter or exit the host while network-based IDS uses data from a network and is scrutinised against a database and it flags those who look suspicious. Audit data from one or several hosts may be used as well to detect signs of intrusions. A significant problem of IDS is how to efficiently divide the normal behaviour and the abnormal behaviour from a huge number of raw information's attributes and how to effectively generate automatic intrusion rules following composed raw data of the network. To accomplish this, different data mining (also known as knowledge discovery in databases) techniques have been studied, like classification, clustering, association and so on can be used to dissect the information.

Data mining techniques have been successfully applied in many different fields including marketing, manufacturing, process control, fraud detection and network management. But they are relatively new approaches for intrusion detection (Chiang and Zhang, 2016; Wurzenberger et al., 2017; Yousif and Hussein, 2014; Omran and Panda, 2016; Khan et al., 2017). Network traffic is massive and information comes from different sources, so the dataset for IDS becomes large. Hence the analysis of data is very difficult in case of large dataset. Data mining techniques are applied on IDS because it can extract the hidden information and deals with large dataset. Presently Data mining techniques play a vital role in IDS. By using data mining techniques, IDS helps to detect abnormal and normal patterns.

The remaining part of the paper is organised as follows: Section 2 describes motivation. Section 3 presents related works on intrusion detection. Section 3 briefly describes the proposed model of our IDS. The performance analysis and evaluation to prove the effectiveness of our model is presented in Section 4. Finally, Section 5 concludes our study and discusses the future works.

2 Motivation

As the experience with deploying IDS for securing computer systems matured, it became obvious that using the knowledge-based IDS (misuse IDS) mechanism alone becomes quite inefficient; with this method is that they require prior knowledge of attack features and hence cannot identify new categories of attack for which signatures have not been developed. Signature database also needs to be updated manually which is generally tedious, expensive as well as time consuming and error prone job. Despite the fact these false positive alarm rates are higher with anomaly, so is its ability to detect new attacks, which are previously unreported; motivates many research efforts to build effective behavioural-based detectors for the purpose of intrusion detection.

IDS which are using either misuse detection alone or anomaly detection are the big drawbacks of current IDS and caused for low detection rate. Higher detection rate depends on the model of the developed detection system. One of the promising solution to this problem is developing hybrid IDS which combines misuse and anomaly detection. Hybrid IDS has become an important solution to detect attacks which have previously encountered and have already stored signatures and new emerging attacks those which have not stored features, imposed by internal and external intruders.

As network traffic or data which is used for analysis by the intrusion detection (ID) is massive, therefore; we need to search a technique that is capable of handling such issues. Data mining techniques have taken beneficial steps towards solutions of different problems in different issues; and IDS is one of them, because of the following reasons. It can process large amount of network traffic and user's subjective evaluation is not necessary and it is more suitable to discover the ignored and hidden information (Luo et al., 2016; Li et al., 2018).

The recent rapid development in data mining contributes to develop wide variety of algorithms suitable for intrusion detection (ID) problems. And most of the literature review shows that, for all practical purposes, many authors applied a single algorithm to address all the attack categories with a record of low performance in many cases. To narrow the gap observed from the preview literature; in this paper, we believe that different algorithms can perform differently on different attack categories, therefore; combining more than one data mining algorithms for intrusion detection can improve the overall detection rate as the drawbacks of one algorithm might be solved by the other one.

There are two main contributions in this research paper. First, we have proposed behavioural-based intrusion detection system (BBIDS) based on data mining approach to detect anomaly as well as misuse intrusions imposed by internal and external intruders. It uses a combination of two data mining techniques, the association rule mining and decision tree. Each decision tree represents a rule set, which categorises data according to the attributes of dataset. After the data is labelled as 'intrusion' and 'normal', the so called 'normal' is given to another data mining algorithm called association rule mining for further classification because decision tree can only classify to those which have stored attributes otherwise it considers the intrusions as 'normal'. Here both types of intrusions, misuse and anomaly can be detected in return false alarm rates will be minimised and the detection rate will be maximised.

Second, we take our work further and implement the system in a platform language then apply knowledge discovery data (KDD) mining dataset for experiment and measure its performance. This encourages further research to improve overall performance of the system, like delay by using parallel programming.

3 Related works

So far, there are some IDS based on data mining techniques that provide promising effects. Zuech et al. (2015) detailed the history and evolution of IDS (Gai et al., 2016; Harrison et al., 2016; Milenkoski et al., 2015). It examined the origins of detecting, analysing and reporting of malicious activity. This paper well described the contribution

of Anderson's paper (He et al., 2016; Elnagdy et al., 2016; Pajouh et al., 2016) published a study outlining ways to improve computer security auditing and surveillance at customer sites and internet of things (IoT). The original idea behind automated ID is often credited to him for his paper on how to use accounting audit files to detect unauthorised access. This ID study paved the way as a form of misuse detection for main frame systems.

Kenkre et al. (2015) and Zarpelão et al. (2017) proposed a network intrusion detection system (NIDS). NIDS detect attacks by observing various network activities. The authors applied one of the efficient data mining algorithms called random forest to build patterns for NID. There are two phases in the framework: offline phase and online phase. The system builds patterns of intrusion in the offline phase and detects intrusions in the online phase. They also discussed the approaches for handling imbalanced intrusions, selecting features and optimising the parameters of random forests. They used KDD'99 datasets for their experimental results. Results showed that the proposed approach provides better performance. Random forest is a decision tree technique which is an ensemble of un-pruned classification or regression trees. Random forest generates many classification trees.

Hodo et al. (2016) and Yousif and Hussein (2014) compared the results of two approaches of IDS (phase and level). Phase consists of three detection phases. The data is input in the first phase which identifies if this record is a normal record or attack. If the record is identified as an attack then the module inputs this record to the second phase which identifies the class of the coming attack. The second phase module passes each attack record according to its class type to phase 3 modules. Phase 3 consists of 4 modules one for each attack class type. Each module is responsible for identifying the attack type of coming record, while the level approach consists of three independent detection levels. The first level is to detect normal and Attack profiles. The second level is to detect normal records and classify the attacks into four categories independently on the results of the first level. The third level is to classify each attack type and normal records. It has done the phases by using decision tree techniques and concluded phase approach has higher classification rate than level approach but it is not clear how was done.

Zuech et al. (2015), Yousif and Hussein (2014) and Al-Yaseen et al. (2017) compared decision tree, naive Bayes and the NBTree (hybrid between decision trees and naïve Bayes) for classifying traffics to either normal or attack by using a standard data set on open source tool. The hybrid algorithm (NBTree) had better predictive power with high accuracy and less error rate than using each algorithm alone but it needs more construction and processing time. Ramos et al. (2017) and Sharma and Gaur (2016) presented a new learning algorithm for anomaly-based network intrusion detection using decision tree in wireless sensor networks(WSN), they adjusted the weights of dataset based on probabilities and split the dataset into sub-dataset until all the sub-dataset belongs to the same class. Here nothing is mentioned whether this kind of technique can classify misuse intrusions (Shidore and Bhusari, 2014; Kumar et al., 2015). Focused on the positive false alarms, which are considered malicious while they are normal in nature. The authors applied genetic algorithm (GP) to identify new intrusions by making use of old signatures of misuse intrusions.

Patel and Aluvalu (2014), Li et al. (2017) and Hamid et al. (2016) aimed to reduce the size of decision tree and thus increase the performance of detection. They mentioned different kinds of pruning techniques and chose reduced error pruning technique. This technique works by evaluating the cost at each decision tree node to determine whether to convert the node into a leaf, prune the left or the right child, or leave the node intact. it proceeds to prune the nodes of a branch as long as both sub-trees of an internal node are pruned and stops immediately if even one sub-tree is kept. This is an important concept which helps to minimise performance time.

Kenkre et al. (2015), Wurzenberger et al. (2017) and Pajouh et al. (2017) made an up-to-date survey on recent studies about NID that was evaluated with standard dataset and then compared ten classifier algorithms on different attack categories. Finally, two models for algorithm selection are proposed with great promise for performance improvement and real-time systems application.

4 Behavioural-based cyber intrusion detection

4.1 System model and assumptions

We have considered a network-based IDS that uses data mining approach to detect anomaly (new) as well as misuse (already known) intrusions imposed by internal and external intruders. The system combines the two types of detections, the misuse and anomaly intrusion detections techniques to make a hybrid system. In misuse detection, each instance in a data set is labelled as 'normal' or 'intrusive' and a learning algorithm is trained over the labelled data. During anomaly detection, normal behaviour of the traffic data is studied and when there is a deviation from the normal behaviour, traffic data is considered as attack.

To explain the proposed behavioural-based cyber intrusion detection (BBCID) system in detail, we have defined two data mining techniques (Al-Yaseen et al., 2015; Chahal and Kaur, 2016; Natesan et al., 2017); Decision tree and association rule mining. As we have explained in detail in the introduction and related work parts, there are several data mining techniques, which are used to design intrusion detection. During the decision tree technique, it predicts the value of the target variable based on several input variables. At each node of the tree, it chooses the attributes of the data that most effectively splits its set of samples into subsets enriched in one class or the other using information gain as splitting criteria. During Association Rule Mining technique, it discovers interesting relations between features in large datasets based on a given thresholds: minimum support (MinSupport) and minimum confidence (MinConfidence). Main features of BBCID system scheme are:

- Hybrid system: it detects attacks which have already previously known features and stored signatures to database tables as well as new attacks where their signatures are not known and stored previously. This helps to increase detection rate and lower false alarms.
- Use of two different algorithms: combining more than one data mining techniques for intrusion detection can improve the overall detection rate as the drawbacks of one technique might be solved by the other one.

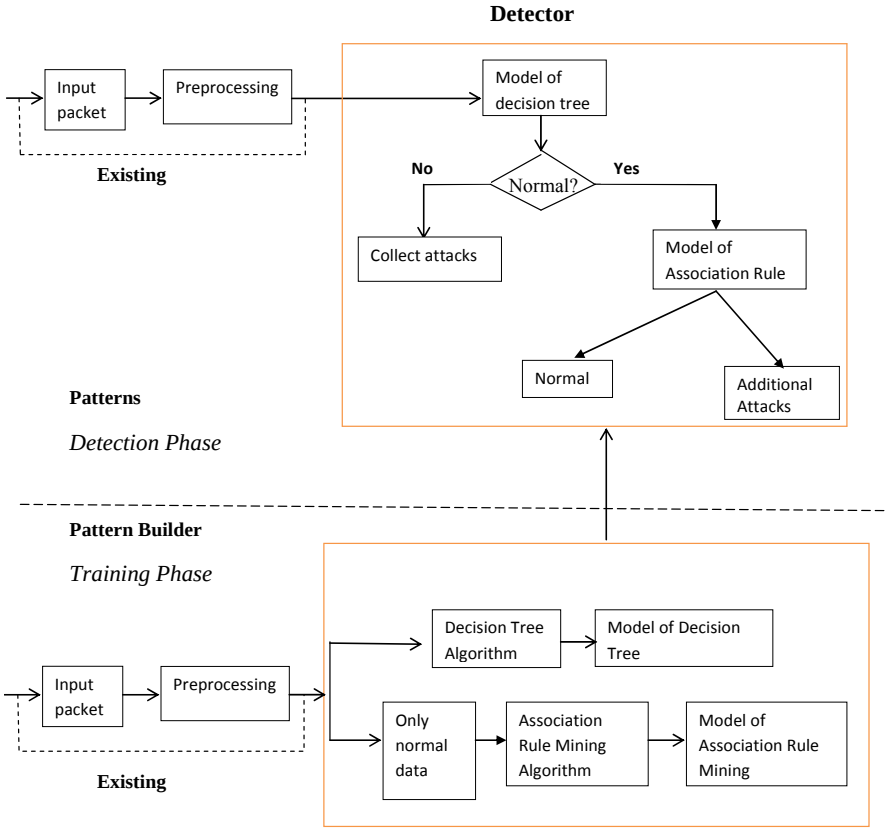
- Use of data mining techniques: can process huge amount of data and it is more useful to find out the ignored and hidden information.
- Use of network-based intrusion detection: monitors packets on the network wire as they pass by some sensor.

We will describe the detailed working principles of the BBCID system in the next section.

4.2 System scheme

Figure 1 shows the proposed BBCID scheme. It uses two data mining techniques and also has two phases which are described below.

Figure 1 Proposed system (BBCID) scheme (see online version for colours)



4.3 Decision tree scheme

In decision tree, the value of the target variable is predicted based on several input variables (Dutt and Borah, 2015). At each node of the tree, attributes of the data that most effectively splits its set of samples into subsets enriched in all class or the other are chosen by using information gain which is a splitting criteria. A decision node specifies a test attribute; an edge is a corresponding to one of the possible attributes values; and leaf, usually named an answer node and it contains the class to which the object belongs. In decision tree, two major phases should be ensured: building the tree based on a given training set; and classification of a new instance.

Information gain (Gain) is given as follows (Yousif and Hussein, 2014; Patel and Rinkal, 2014; Jow et al., 2017; Miller et al., 2015):

$$Entropy: H\left(p_1, p_2, \ldots, p_s\right) = \sum_{i=1}^{s} (pi \log(1/pi)) \tag{1}$$

Given a data set D, $H(D)$ finds the amount of entropy in class-based subsets of the data set. When that subset is split into s new subsets $S = \{D_1, D_2, \ldots, D_s\}$ using some attribute, we can again look at the entropy of those subsets.

$$Gain(D, S) = H(D) - \sum_{i=1}^{s} p(Di)H(Di) \tag{2}$$

$$Gain\ ratio = Gain(D, S) \frac{Gain(D, S)}{H\left(\frac{|D_1|}{|D|}, \ldots, \frac{|D_s|}{|D|}\right)} \tag{3}$$

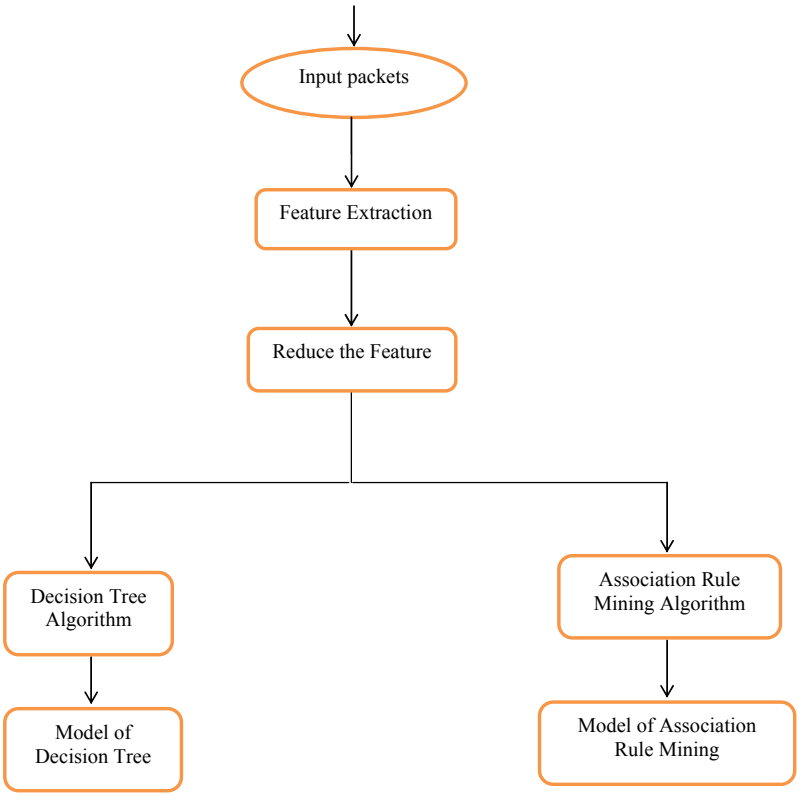
4.4 Association rule mining scheme

Association rule mining discovers interesting relations between features in large datasets. It searches a frequently occurring item set from a large dataset. It works in two forms: Frequent Item set Generation, generates all set of items whose support is greater than the specified threshold and Association Rule Generation, it generates the association rules in the form of if-then statements that have confidence greater than the specified threshold using the previously generated frequent item sets.

4.5 Training phase scheme

Input network dataset is separated for training and testing phases. In training phase, as Figure 2 shows, the first step is to perform the pre-processing on the input data, that includes feature extraction and feature reduction. Both data mining algorithms are trained on the same data. After decision tree algorithm is trained on the given data set, its model is built. And the same procedure is applied to association rule mining algorithm too; when it is trained with the same given data set, the association rule mining model is developed. This is a pattern builder phase, which is going to be used in the intrusion (testing) phase.

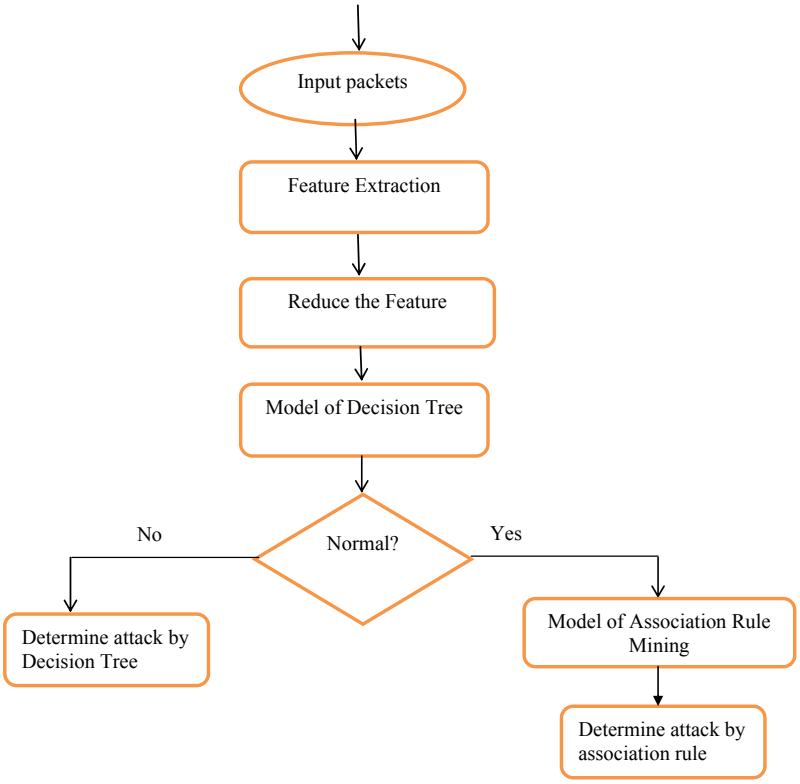
Figure 2 Training phase scheme (see online version for colours)



4.6 Intrusion phase scheme

In intrusion phase, also it is called testing phase as shown in Figure 3, decision tree would be the basic classifier, it classifies the known attacks based on the rules or patterns that are already produced in the training phase. The pre-processed data would be given to decision tree first, if the data is classified into attack or intrusion, it could be determined the kind of attack by decision tree. On the other hand, if the data is classified as normal, the so called normal is collected and input to association rule mining algorithm for further classification because decision tree can only classify known attacks, those have stored features. Otherwise, this decision tree algorithm considers the intrusion as 'normal'.

Figure 3 Testing phase scheme (see online version for colours)



5 Implementation, performance analysis and evaluation

5.1 Experimental setup

In order to implement and evaluate the proposed model for network intrusion detection, all experiments were performed using a hardware specification of Intel Core i5-5200U CPU @2.20GHz processor with 4.00GB of RAM. And software specification of Windows 7, 32-bit operating system. We have developed our system using java programming language on Net beans IDE 7.4 and Microsoft SQL Server database for storing the features of the attack and normal data.

5.2 System dataset

As authors (Günes et al., 2005) mentioned in 1999, during the international knowledge discovery and data mining tools competition, the knowledge discovery and data (KDD'99) mining was developed by the Massachusetts Institute of Technology (MIT) and the name KDD'99 is given after that. The original Transmission Control Protocol (TCP) dump files were pre-processed for utilisation in the IDS benchmark. In order to do

so, packet information in the TCP dump file was summarised into connections. Connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source internet protocol (IP) address to a target IP address under some well-defined protocol that results in 41 features for each connection.

Each TCP connection was labelled as 'normal' or 'attack' with a specific attack type. The attack types are: DOS, U2R, R2L and PROBE. And features are grouped into four categories:

- Basic features: features that identify packet header properties which represent connection critical metrics.
- Content features: features represent useful information extracted from the packets that help experts to identify known forms of attacks.
- Time-based traffic features: features that are computed with respect to a two-seconds time interval window. These could be divided into two groups, same host features and same service features.
- Host-based traffic features: features that are computed with respect to a connection window of 100 connections. Statistics are calculated from a historical data that is estimated from the last hundred used connections to the same destination address. These are useful to detect slow probing attacks that scan hosts or ports using at much larger time interval than 2 seconds.

Even though, KDD'99 is one of the most popular benchmark datasets used to choose proper intrusion detection metrics, Ayman et al. (2014), Kumar et al. (2015) and Budgaga et al. (2017) mentioned important drawback that is having large number of redundant records that could bias learning algorithm to the classes with large repeated records. NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set which are mentioned in Ayman et al. (2014), Korczynski et al. (2016), Midi et al. (2017) and Bostani and Sheikhan (2017). It does not include redundant records; therefore, the classifiers will not be biased towards more frequent records. NSL-KDD is a reduced version of the complete KDD'99 dataset which have the same features.

5.3 Feature reduction

The features in NSL-KDD intrusion detection dataset are large in number; they are 41 features in each connection as mentioned in the above. This has a disadvantage of poor resource utilisation, high computational cost and so on and leads to overall low system performance. Not all features in NSL-KDD intrusion detection dataset are equally important. Certain features have no relevance or contribution to detect any intrusion attack type. Some features are important to detect all attack types and certain features are important to detect certain attack types. Therefore, it is advisable to reduce some features to improve the overall performance of the system.

So far, different papers are proposed on various feature selection methods. Omran and Panda (2016), Ayman et al. (2014), Moamed and Helmi (2016) and Mayzaud et al. (2016) presented some important model of relevance feature selection. Günes et al. (2015), Wei et al. (2016) and Wang et al. (2016a) used information gain to select the relevant feature. Information gain of all the 41 features is calculated, therefore; a feature with the highest information gain would be the most discriminating feature for each class.

Ayman et al. (2014) used gradually add feature and gradually remove feature method. By continuously testing on selected IDS algorithms, authors concluded that, the best features set that achieved an optimised detection performance and accuracy compared to the full 41 features set was the following 11 features set: duration, service, flag, source_bytes, destination_bytes, hot, root_shell, count, serror_rate, diff_srv_rate, dest_hos_ diff_srv_rate. We have adopted the 11 features set from papers (Ahmed, 2017; Ayman et al., 2014; Jabbar et al., 2017; Ji, et al., 2016), because it is tested and evaluated by different IDS algorithms and it showed high detection rate in all algorithms in comparison with the other two papers (Jabber et al., 2017; Omran and Panda, 2016; Wang et al., 2016b) which we have reviewed.

5.4 System implementation

We have implemented the model using java programming language on net beans tool and SQL server database is used to store the records. We have used a standard dataset, NSL-KDD IDS dataset by reducing the features from 41 to 11 using a technique in Omran and Panda (2016), Ayman et al. (2014) and Osanaiye et al. (2016), so that performance could be improved. Total number of features would become 11. Eleven are main features and one is the classifier feature called 'class', which is a dependent variable. NSL-KDD contains 125,973 records. We have separated the dataset into two parts, training and testing. During the training, patterns or rules are produced. And during testing, identification of intrusions (attacks) and normal data are taken place.

In order to show the impact of training dataset size, we have taken percentage split and ten-fold cross validation techniques for separating NSL_KDD datasets into training and testing

Table 1 Percentage split of NSL_KDD datasets into training and testing

Total dataset records	*Percentage split*	*Training*	*Testing*
125,973	90% to 10%	113,376	12,597
	80% to 20%	100,778	25,195
	70% to 30%	88,181	37,792

Table 2 Ten-fold cross validation NSL_KDD datasets into training and testing

Total dataset records	*10 fold cross validation*	*Training*	*Testing*
125,973	Fold-1	113,375	12,598
	Fold-2	113,375	12,598
	Fold-3	113,375	12,598
	Fold-4	113,376	12,597
	Fold-5	113,376	12,597
	Fold-6	113,376	12,597
	Fold-7	113,376	12,597
	Fold-8	113,376	12,597
	Fold-9	113,376	12,597
	Fold-10	113,376	12,597

Table 1 and Table 2 show selected dataset records which are used in our development, separated into training and testing datasets. In our implementation:

1 First option, we have used 90% of the complete dataset for training and the remaining 10% for the testing.

2 Second option, we have used 80% of the total dataset for our training and the remaining 20% of the total dataset for testing the model.

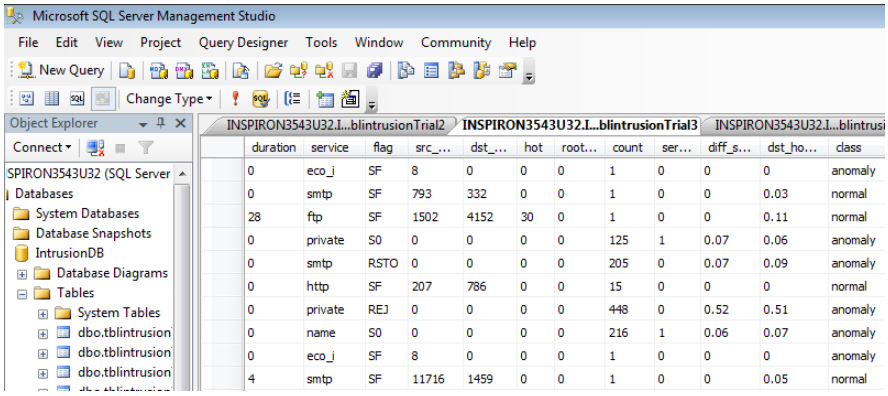
3 Third option, we have taken 70% of the total dataset for training to build the model and the remaining 30% would be used for the testing of the model.

4 And the fourth and final option, we have trained and tested using the ten-fold cross validation testing option.

The total dataset is randomly partitioned into ten sub-datasets. Of the ten sub-datasets, a single sub-dataset is retained as the validation data for testing the model and the remaining nine sub-datasets as training data.

We have recorded all of the above datasets into our SQL server database tables for later use of rule generation and intrusion parts.

Figure 4 Parts of the database files and their NSL_KDD dataset records for the system (see online version for colours)



We could summarise our detection model procedures by the following steps:

1 collected input network datasets are separated for training and testing phases

2 perform pre-processing of both training and testing datasets

3 train both decision tree and association rule mining algorithms so that the patterns or rules would be produced

4 decision tree would be selected as base classifier, therefore, it classifies the known attacks based on the database of its features (attack and normal), known attacks are classified and collect the output

5 normal outputs would be collected from decision tree and given as input to association rule mining so that new attacks would be detected

6 at last, outputs from both algorithms would be collected for analysis

7 finally, analysis and interpretation of results would be taken place.

5.5 Decision tree model implementation

In order to implement the decision tree scheme, we have to come up with an algorithm that can wisely solve the problem. C4.5 is very powerful and popular decision tree algorithm for decision-making and classification problem whose output is a tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification and for this reason, C4.5 is often referred to as a statistical classifier.

The pseudo code for building C4.5 decision trees is written below (Yousif and Hussein, 2014):

1 check for a base case

2 for each attribute find the normalised information gain ratio from (3).

3 let a_best be the attribute with the highest normalised information gain

4 create a decision node that splits on a_best

5 repeat on the sublists obtained by splitting on a_best. Add the obtained nodes as children of the a_bestnode

Decision tree algorithms use the strategy of future generations, from root to leaves. To ensure this process, the attribute selection measure is used, taking into account the discriminative power of each attribute over the classes in order to choose the 'best' one as the root of the (sub) decision tree. In other words, best attribute should be used as a root node for splitting the tree. Objective criteria for judging the efficiency of the split is needed and information gain measure is used to select the test attribute at each node in the tree. The attribute with the highest information gain is chosen as the test attribute for the current node. This attribute minimises the information needed to classify samples in the resulting partitions.

By fetching each training dataset from SQL server database tables, we have written a java program to produce decision rules using C4.5 decision tree algorithm. Then we have written the rules produced above into java code as' if –then' to classify the test dataset. Finally, by retrieving the test dataset from the database tables, we have classified in to attack and normal data. The normal data are collected and stored for further classification using the next algorithm, Association rule mining algorithm.

5.6 Association rule mining model implementation

From decision tree algorithm, of C4.5, we have collected normal data for further classification. C4.5 is mostly used for known attacks, so we need to come up with another algorithm which can deal with new attacks.

The Apriori algorithm is one of the most influential mining association rule algorithm and the rule is expressed by frequent item collection. The association rule has two important attributes: Support level P(XUY), namely the probability of the two items of collections X and Y which simultaneously appear in the database table transactions.

Confidence level P(Y|X), namely probability that collection X appears in items of database table transactions, items of collection Y also simultaneously display. The rules which simultaneously satisfy the smallest support threshold value and the smallest confidence level threshold value are called the strong rule. Giving transactions, the association rule mining creates the rules whose support and confidence level is bigger separately than the smallest support and confidence level which the user assigns. The Apriori core algorithm has used the recursion method in order to produce all frequency collections.

- *Support*: The support, *supp*(*X*) of an itemset *X* is defined as the proportion of transactions in the data set which contain the itemset.

$$supp(X) = \frac{No.\ of\ \ transcations\ which\ contain\ the\ itemseX}{Total\ no.\ of\ \ transcations}$$

- *Confidence*: The confidence of a rule is defined:

$$Conf(X \rightarrow Y) = Supp(XUY) \Big/ Supp(X) \tag{4}$$

Association rule generation is usually contains two separate steps: First, minimum support is applied to find all frequent itemsets in a database. And second, these frequent itemsets and the minimum confidence constraint are used to form rules.

Below is the Apriori algorithm pseudo code:

```
procedure Apriori (T, minSupport)
{//T is the database and minSupport is the minimum support
L1= {frequent items};
for (k = 2; Lk-1 !=; k++) {
Ck = candidates generated from Lk-1
//that is cartesian product Lk-1 x Lk-1 and eliminating any k-1 size itemset that is not frequent
for each transaction t in database do{
#increment the count of all candidates in Ck that are contained in t
Lk = candidates in Ck with minSupport
}//end for each
}//end for
return UkLk;
}
```

By feeding the normal features, we have implemented Apriori algorithm to produce association rules in java using an open source data mining library, called SPMF. We have chosen a convenient thresholds of Support and Confidence; 30% and 80% respectively.

Then after, we have written the rules to java code to further detect the output of decision tree. Having collected the normal NSL-KDD testing dataset from database (C4.5 normal output), association rule mining model has detected further attacks.

6 Performance evaluation and results

To evaluate our detection system, we applied the evaluation criteria as follows:

- Detection rate (DR): it is the percentage of normal and attack data are classified correctly from the given number of total dataset records.
- False positive (FP): it is when normal connections are incorrectly classified as intrusions or attacks.
- False negative: this is when intrusions are mistakenly identified as normal.
- False alarm rate: this is the inverse of detection rate, when number of normal and attack data are classified incorrectly.

As it is shown in Table 1, the total NSL_KDD dataset is divided into three sample ratios for the evaluation of our system, these are:

1 90% to 10%: 90% of the total dataset for training and the remaining 10% is for testing

2 80% to 20%: 80% of the total dataset for training and the remaining 20% is for testing

3 70% to 30%: this is also shows, 70% of the total dataset to train the system and the 30% to test the system

Table 3 confusion matrix

Standard metrics		*Prediction connection label*	
		Normal	*Intrusions (attacks)*
Actual connection label	Normal	True negative	False alarm
	Intrusion (attacks)	False negative	Correctly detected attacks

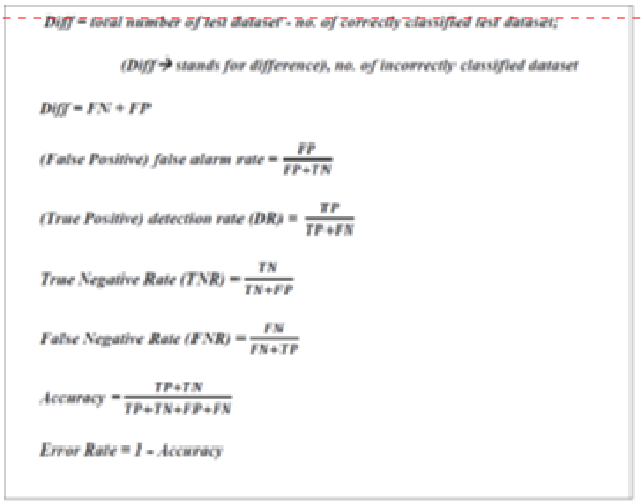
As it is described in the implementation part above, our system is evaluated on all the three NSL_KDD dataset ratios step by step.

First we implemented the decision tree algorithm and apply on these three sample dataset ratios. We have derived the following formula. The system calculates all correctly classified data, number of unclassified data, detection rate and false alarm rate.

Testing dataset and diff (incorrectly classified as normal and attack) are given to calculate detection rates, false alarm rates, accuracy. It is clear that detection rate is the inverse of false alarm rate. Based on the above formula, we have evaluated the two algorithms (decision tree and association rule mining) independently and then the combined approach (our hybrid model). This approach helps us to compare the performance of each algorithm alone to the combined hybrid model.

In all the three approaches, we have performed the four experiments to learn the impacts of various training dataset sizes.

Figure 5 Decision tree algorithm



Comment [a1]: Author: Please provide a clear version of this figure. Preferably an EPS file.

The first classification model that we used was the decision tree. The results of this algorithm are summarised in Tables 4 and 5. In Table 4, we feed total NSL_KDD datasets of 125,973 records to decision tree algorithm. We have performed three percentage split of training to testing ratios in order to observe the impact of training dataset size on results. Results show us size of the training has little impact on detection rate and accuracy of the model in this stage.

Table 4 Results of decision tree algorithm classification model

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
90%–10%	5,835	6,754	1	7	99.880	0.015	99.936	0.064
80%–20%	11,733	13,437	11	14	99.881	0.082	99.901	0.099
70%–30%	17,549	20,224	19	22	99.875	0.094	99.892	0.108

We can see that, when number of training dataset decreases and testing dataset increases, it is obvious that the detection rate declines and false alarm rate raises in contrary. That is because, when we make our system learn in large dataset, it can get a chance to build almost all of the patterns or rules of the features; therefore, it could handle any new feature encountered in the testing dataset. After we have done the identification of attacks and normal dataset using decision tree algorithm, we have collected the so called normal data which is identified by C4.5 algorithm and feed as testing data for Association Rule Mining (apriori algorithm) for further classification. As it is mentioned earlier, Decision tree algorithms almost classifies known attacks, whose signatures are already known and stored to database otherwise it considers the new attack as normal. On the other hand, the anomaly detection that we have used Association Rule Mining (Apriori algorithm), detects new attacks, those who have not stored signatures but it also considers some normal data as attack, as it works by learning the current normal behaviour and when it

deviates from this normal behaviour it considers as attack. As behaviours are not constant and always changing, sometimes anomaly detector considers the old normal behaviour as attack. Therefore, combining these techniques compensate the drawbacks of one to another.

In Table 5, we have used ten-fold cross validation testing option as an alternative to percentage split. We have divided the total NSL_KDD dataset randomly into ten parts. Decision tree is trained on the 10–1 folds and validated on the remaining 1 fold data. Then the performance measure is reported as an average of the values computed in the loop.

Table 5 Results of Decision tree algorithm classification model (ten-fold cross validation)

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
Fold-1	5,895	6,688	2	13	99.78	0.030	99.881	0.119
Fold-2	5,823	6,753	9	13	99.777	0.133	99.825	0.175
Fold-3	5,886	6,697	8	7	99.881	0.119	99.881	0.119
Fold-4	5,998	6,589	4	6	99.900	0.061	99.921	0.079
Fold-5	5,895	6,690	8	4	99.932	0.119	99.905	0.095
Fold-6	5,753	6,829	9	6	99.896	0.132	99.881	0.119
Fold-7	5,750	6,834	2	11	99.809	0.029	99.897	0.1032
Fold-8	5,817	6,765	8	7	99.880	0.118	99.881	0.119
Fold-9	5,896	6,684	9	8	99.864	0.134	99.865	0.135
Fold-10	5,835	6,754	1	7	99.880	0.015	99.936	0.064
Average					99.859	0.089	99.887	0.113

The second classification model that we used was the association rule mining. The results of this algorithm are summarised in Tables 6 and 7. We trained the algorithm (association rule mining algorithm) on normal part of NSL_KDD datasets. For example, in the first line of Table 11, we collected the normal part of the 90% the total NSL_KDD dataset for training the algorithm and the remaining 10% (whole 10%) is used for testing. The model works by comparing the incoming testing datasets to normal behaviour of the system. When there is a deviation it considers as an attack.

Table 6 Results of Association rule mining classification model (percentage split)

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
90%–10%	4,181	6,748	7	1,661	71.56	0.104	86.76	13.24
80%–20%	8,457	13,431	17	3,290	71.99	0.126	86.87	13.13
70%–30%	12,658	20,200	21	4,913	72.04	0.104	86.94	13.06

The third classification model that we used was the hybrid algorithm with decision tree and association rule mining. The results of this model are summarised in Table 8 and Table 9.

Table 7 Results of Association Rule Algorithm Classification model (ten-fold cross validation)

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
Fold-1	4,301	6,680	10	1,607	72.8	0.15	87.17	12.84
Fold-2	4,211	6,751	11	1,625	72.16	0.16	87.01	12.99
Fold-3	4,283	6,700	5	1,610	72.68	0.08	87.18	12.82
Fold-4	4,334	6,591	2	1,670	72.19	0.03	86.73	13.27
Fold-5	4,237	6,686	12	1,662	71.83	0.18	86.71	13.29
Fold-6	4,197	6,830	8	1,562	72.88	0.12	87.54	12.46
Fold-7	4,187	6,832	4	1,574	72.68	0.06	87.47	12.53
Fold-8	4,202	6,769	4	1,622	72.15	0.06	87.09	12.91
Fold-9	4,275	6,683	10	1,629	72.41	0.15	86.99	13.01
Fold-10	4,181	6,748	7	1,661	71.56	0.10	86.76	13.24
Average					72.33	0.106	87.06	12.94

Table 8 Results of Hybrid classification model (percentage classification)

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
90%–10%	5,835	6,747	8	7	99.88	0.12	99.88	0.12
80%–20%	11,730	13,421	31	13	99.89	0.23	99.83	0.18
70%–30%	17,553	20,183	36	20	99.89	0.18	99.85	0.15

Our hybrid model trained on normal part of the NSL_KDD datasets. After we have done the identification of attacks and normal dataset using decision tree algorithm, we have collected the so called normal data output. Then we feed normal output data to association rule mining (Apriori algorithm) as our testing data. Association rule mining identifies further classification into attack and normal. As it is mentioned earlier, decision tree algorithm almost classifies known attacks; whose signatures are already known and stored to database otherwise it considers the new attack as normal. On the other hand, association rule mining (Apriori algorithm) detects new attacks, those which have not yet stored signatures and considers some normal data as attack. We mentioned earlier that association rule mining works by letting learn on the current normal behaviour and when there is a deviation from the normal behaviour it considers as an attack. Behaviours are not constant and always changing. Sometimes anomaly detector considers the old normal behaviour as attack. Therefore, by combining both techniques, the drawbacks of each other will be complemented.

Our system implementation calculates and displays TP, TN and diff (sum of FP and FN) from the identified data as attack and normal. Based on the formula given in equation (4), we have calculated detection rate, accuracy, false alarm rate and error rate for all the three classification models.

Tables 10 and 11 show the identification of attack and normal data using our hybrid model.

Table 9 Results of hybrid classification model (ten-fold cross validation)

Training-to-testing	*TP*	*TN*	*Diff*		*DR (%)*	*FAR (%)*	*Accuracy (%)*	*Error rate*
			FP	*FN*				
Fold-1	5,896	6,679	11	12	99.78	0.16	99.82	0.18
Fold-2	5,823	6,746	16	13	99.78	0.24	99.77	0.23
Fold-3	5,887	6,693	12	6	99.89	0.18	99.86	0.14
Fold-4	5,998	6,587	6	6	99.90	0.09	99.91	0.09
Fold-5	5,896	6,682	16	3	99.95	0.24	99.85	0.15
Fold-6	5,753	6,822	16	6	99.89	0.23	99.83	0.18
Fold-7	5,750	6,831	5	11	99.81	0.07	99.87	0.13
Fold-8	5,817	6,763	10	7	99.88	0.15	99.87	0.14
Fold-9	5,896	6,675	18	8	99.86	0.27	99.79	0.21
Fold-10	5,835	6,747	8	7	99.88	0.12	99.88	0.12
Average					99.87	0.175	99.743	0.156

In Table 8, in the first row, for our training, we have used only normal data of 90% of the total NSL_KDD dataset and for our testing, the normal output data of decision tree of its 10% testing dataset. Our system calculates the difference (diff). As we have shown above, diff is the sum of false positive and false negative which is calculated by subtracting total correctly classified data from given total testing data. True positive (total data which is identified as attack minus false positive) and true negative (total data which is identified as normal minus false negative) are calculated. By taking false positive, false negative, true positive and true negative parameters, detection rate, false alarm rate, accuracy and error rate are derived as given in equation (4). We have got a detection rate of 99.880% which is similar with decision tree model but higher than association rule mining model. In the second row, we have used the normal data of 80% of the total NSL_KDD dataset for training and the normal output data of decision tree 20% testing dataset for testing. We have got a detection rate of 99.889%. This is higher than both classification models. In the third row, we have used 70% of the total NSL_KDD dataset for training the model and normal output data of 10% testing dataset from decision tree for testing. 99.886% detection rate which is higher than both models is produced.

In Table 9, results are produced using ten-fold cross validation testing option. We have achieved 99.865% of detection rate. This is higher than both classification models.

In terms of detection rate, our hybrid model is better than both classification models. It is slightly higher than decision tree classification model and totally exceeds association rule mining classification model. But in terms of accuracy, the hybrid model is slightly lower than decision tree classification model and still exceeds the association rule mining model.

Now, as we have explained above, both misuse and anomaly techniques for detection system have their own drawbacks and advantages. And combining them as a hybrid system could compensate each other, so that, our hybrid model's performance would be raised.

Here as we shown results of the two algorithms, decision tree and association rule mining in Table 4 to 7 respectively. We have trained decision tree algorithm on the given NSL_KDD dataset samples, then classification of attack and normal is taken place by

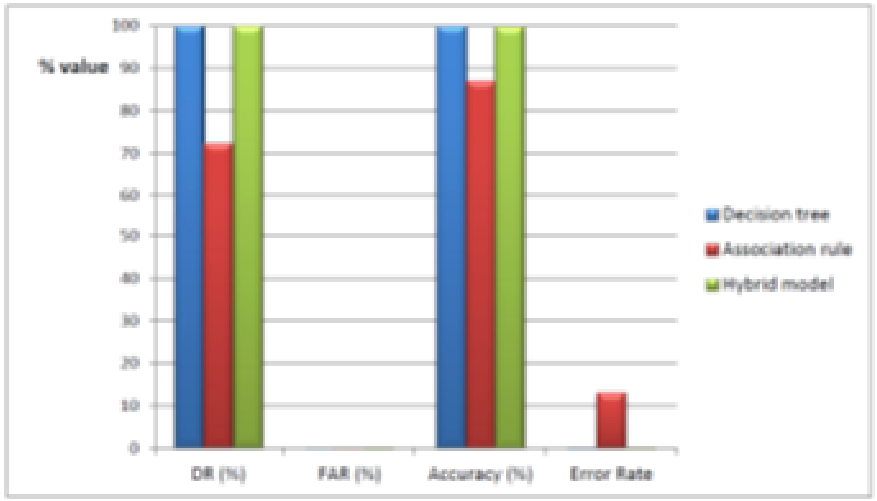
feeding the test data. After all the datasets are classified, we have collected all the normal datasets and given to association rule mining as a test dataset. Association rule mining is trained on all normal datasets which are collected from the total NSL_KDD dataset. We have considered those all normal NSL_KDD datasets as normal working behaviours of systems. Now, we have got the advantages of both algorithms. Decision tree algorithm is better at detecting known attacks but suffers from false negatives, on the other hand, advantages of both algorithms. Decision tree algorithm is better at detecting known attacks but suffers from false negatives, on the other hand, association rule mining algorithm is better at detecting new attacks but suffers from false positives.

Therefore, by combining both algorithms as hybrid, we have obtained results given in Table 8 and Table 9. Detection rate, accuracy, false alarm rate, error rate and other parameters are displayed.

As we can see from all the results above, our hybrid model has better detection rate than both classification models. But it has slightly less accuracy than decision tree classification model but still far higher accuracy than association rule mining classification model.

Figure 5 is showing a comparison between the three classifier models in a graphical way. The figure clearly indicates that in terms of detection rate the hybrid model classifier is better than both models. The hybrid model's detection rate is slightly better than the decision tree model and far better than association rule mining model. Decision tree classification model is the next. In case of accuracy, decision tree classification model is better than both models and hybrid model is the next.

Figure 6 Classifiers comparison chart (see online version for colours)



Comment [a2]: 1)Author: Please note that we cannot renumber this as Figure 5, as Figure 5 already exist above.

2) Author: Please provide a clear version of this figure. Preferably an EPS file.

7 Conclusions and future works

In this thesis work, a hybrid IDS, called behavioural-based cyber network intrusion detection system (BBCNIDS) is proposed based on data mining techniques of decision tree and association rule mining. It combines simultaneously the misuse and anomaly intrusion detections. The main feature of BBCID system is that the application of hybrid system that detects attacks which have previously known and stored features as well as new attacks whose signatures are neither known nor stored previously. This helps to increase detection rate and lower false alarms. Use of two different algorithms; combining more than one data mining techniques for intrusion detection can improve the overall detection rate as the drawbacks of one technique might be solved by the other one. Use of data mining techniques, this can process huge amount of data and it is more useful to find out the ignored and hidden information. Use of network-based intrusion detection, monitors packets on the network wire as they pass by some sensor.

We further implemented the model in java programming language by taking NSL_KDD dataset. The total NSL_KDD dataset is separated for training and testing in the ratio of 90% to 10%, 80% to 20% and 70% to 30%. We have also used ten-fold cross validation testing option. Making the model learn during the training phase and classify attacks during the intrusion phase (testing phase). All the three classification models' detection rate, accuracy, false alarm rate, error rate and other parameters are calculated and displayed in respected tables.

Our first algorithm (DT) produces detection rates of 99.880%, 99.881% and 99.875% using the three percentage split given in Table 6 respectively. And 99.8599% of detection rate using the ten-fold cross validation given in Table 7.

From the association rule mining algorithm alone, we have got a detection rate of 71.557%, 71.993% and 72.039% using the percentage split given in Table 6 respectively. And 72.3328% of detection rate using the ten-fold cross validation from Table 7.

Our hybrid model's classification results are far better than association rule mining and slightly higher than decision tree algorithm. 99.880%, 99.889% and 99.886% detection rates are achieved using percentage split given in Table 6 respectively. And 99.865% of detection rate using ten-fold cross validation from Table 7. Our hybrid model detection rates exceed both algorithms. But in the case of accuracy, our hybrid model is slightly less than decision tree model and higher than association rule mining model.

Further classification of attack types to DOS, U2R, R2L and PROBE are not taken place during this stage. NSL_KDD dataset does not have specific attack types. NSL_KDD dataset contains only classes 'anomaly' and 'normal'.

In our future work, the system will be deployed in real environment and evaluated using real-time traffic data. We will also enhance the system using parallel programming in order to suppress the computation delay encountered in this stage.

References

Ahmed, T. (2017) 'An effective approach of detecting DDoS using artificial neural networks', 2017 *International Conference on Wireless Communications; Signal Processing and Networking (WiSPNET)*, pp.707–711.

Airehrour, D., Gutierrez, J. and Ray, S.K. (2016) 'Secure routing for internet of things: a survey', *Journal of Network and Computer Applications*, Vol. 66, pp.198–213.

Al-Jarrah, O.Y., Omar A., Yoo, P.D., Muhaidat, S., Taha, K. and Kwangjo, K. (2016) 'Data randomization and cluster-based partitioning for botnet intrusion detection', *IEEE Transactions on Cybernetics*, Vol. 46, No. 8, pp.1796–1806.

Alrawais, A., Alhothaily, A., Hu, C. and Cheng, X. (2017) 'Fog computing for the internet of things: Security and privacy issues', *IEEE Internet Computing*, Vol. 21, No. 2, pp.34–42.

Al-Yaseen, W.L., Othman, Z.A. and Nazri, M.Z.A. (2017) 'Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system', *Expert Systems with Applications*, Vol. 67, pp.296–303.

Al-Yaseen, W.L., Othman, Z.A., Nazri, A. and Zakree, M. (2015) 'Hybrid modified-means with C4. 5 for intrusion detection systems in multiagent systems', *The Scientific World Journal*.

Arrington, B., Barnett, L., Rufus, R. and Esterline, A. (2016) 'Behavioral modeling intrusion detection system (BMIDS) using internet of things (IoT) behavior-based anomaly detection via immunity-inspired algorithms', in 25th *International Conference on Computer Communication and Networks (ICCCN)*, IEEE, pp.1–6.

Ayman I., Amr, M. Gody, T. and Barakat, M. (2014) 'Relevant feature selection model using data mining for intrusion detection system', *International Journal of Engineering Trends and Technology (IJETT)*, Vol. 9, No. 10, pp.2231–5381.

Bertino, E., Choo, K-K.R., Georgakopolous, D. and Nepal, S. (2016) 'Internet of things (IoT): smart and secure service delivery', *ACM Transactions on Internet Technology (TOIT)*, Vol. 16, No. 4, p.22.

Bostani, H. and Sheikhan, M. (2017) 'Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept', *Pattern Recognition*, Vol. 62, pp.56–72.

Budgaga, W., Malensek, M., Pallickara, S.L. and Pallickara, S. (2017) 'A framework for scalable real-time anomaly detection over voluminous, geospatial data streams', *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 12.

Chahal, J.K. and Kaur, J. (2016) 'Use of data mining techniques in intrusion detection – a survey', *Imperial Journal of Interdisciplinary Research*, Vol. 2, No. 6.

Chiang, M. and Zhang, T. (2016) 'Fog and IoT: an overview of research opportunities', *IEEE Internet of Things Journal*, Vol. 3, No. 6, pp.854–864.

Chiche, A. and Meshesha, M. (2017) 'Constructing a predictive model for an intelligent network intrusion detection', *International Journal of Computer Science and Information Security*, Vol. 15, No. 3, p.392.

Dutt, I. and Borah, S. (2015) 'Some studies in intrusion detection using data mining techniques', *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 4, No. 7.

Elnagdy, S.A., Qiu, M. and Gai, K. (2016) 'Understanding taxonomy of cyber risks for cybersecurity insurance of financial industry in cloud computing, *3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, pp.295–300.

Ezzat, H., Badr, S.M. and Shaheen, M.A. (2012) 'Phases vs. levels using decision trees for intrusion detection systems', *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 10, No. 8.

Gai, K., Qiu, M., Tao, L. and Zhu, Y. (2016) 'Intrusion detection techniques for mobile cloud computing in heterogeneous 5G', *Security and Communication Networks*, Vol. 9, No. 16, pp.3049–3058.

Gendreau, A.A. and Moorman, M. (2016) 'Survey of intrusion detection systems towards an end to end secure internet of things', in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, IEEE, pp.84–90.

Günes, A., Zincir-Heywood, N. and Heywood, M.I. (2015) 'Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets', in *Proceedings of the Third Annual Conference On Privacy, Security and Trust*.

Hamid, Y., Sugumaran, M. and Balasaraswathi, V.R. (2016) 'Ids using machine learning-current state of art and future directions', *British Journal of Applied Science and Technology*, Vol. 15, No. 3.

Harrison, D.C., Seah, W.K.G. and Rayudu, R. (2016) 'Rare event detection and propagation in wireless sensor networks', *ACM Computing Surveys (CSUR)*, Vol. 48, No. 4, p.58.

He, H., Maple, M., Watson, T., Tiwari, A., Mehnen, J., Jin, Y. and Gabrys, B. (2016) 'The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing and other computational intelligence', in *IEEE Congress on Evolutionary Computation (CEC)*, IEEE, pp.1015–1021.

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P-L, Iorkyase, E., Tachtatzis, C. and Atkinson, A. (2016) 'Threat analysis of IoT networks using artificial neural network intrusion detection system', in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, pp.1–6.

Jabbar, M.A., Aluvalu, R. and Reddy, S. (2017) 'Cluster based ensemble classification for intrusion detection system', in *Proceedings of the 9th International Conference on Machine Learning and Computing*, ACM, pp.253–257.

Ji, S-Y., Jeong, B.K., Choi, S. and Jeong, D.H. (2016) 'A multi-level intrusion detection method for abnormal network behaviors', *Journal of Network and Computer Applications*, Vol. 62, pp.9–17.

Jow, J., Xiao, Y. and Han, W. (2017) 'A survey of intrusion detection systems in smart grid', *International Journal of Sensor Networks*, Vol. 23, No. 3, pp.170–186.

Kenkre, P.S., Pai, A. and Colaco, L. (2015) 'Real time intrusion detection and prevention system', *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, *Advances in Intelligent Systems and Computing*, Springer, Vol. 327, p.201.

Khan, F.A., Imran, M., Abbas, H. and Durad, M.H. (2017) 'A detection and prevention system against collaborative attacks in mobile ad hoc networks', *Future Generation Computer Systems*, Vol. 68, pp416–427.

Korczynski, M., Hamieh, A., Huh, J.H., Holm, H., Rajagopalan, S.R. and Fefferman, N.H. (2016) 'Hive oversight for network intrusion early warning using DIAMoND: a bee-inspired method for fully distributed cyber defense', *IEEE Communications Magazine*, Vol. 54, No. 6, pp.60–67.

Kumar, G. and Kumar, K. (2015) 'A multi-objective genetic algorithm based approach for effective intrusion detection using neural networks', in *Intelligent Methods for Cyber Warfare*, pp.173–200, Springer, Cham.

Kumar, N., Singh, J.P., Bali, R.S., Misra, S. and Ullah, S. (2015) 'An intelligent clustering scheme for distributed intrusion detection in vehicular cloud computing', *Cluster Computing*, Vol. 18, No. 3, pp.1263–1283.

Leloglu, E. (2016) 'A review of security concerns in internet of things', *Journal of Computer and Communications*, Vol. 5, No. 1, p.121.

Li, B., Lu, R., Wang, W. and Choo, K-K.R. (2017) 'Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system', *Journal of Parallel and Distributed Computing*, Vol. 103, pp.32–41.

Li, L., Yu, Y., Bai, S., Hou, Y. and Chen, X. (2018) 'An effective two-step intrusion detection approach based on binary classification and k-NN', *IEEE Access*, Vol. 6, pp.12060–12073.

Luo, G, Wen, Y. and Lingyun, X. (2016) 'Network attack classification and recognition using hmm and improved evidence theory', *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 4.

Mayzaud, A., Sehgal, A., Badonnel, R., Chrisment, I. and Schönwälder, J. (2016) 'Using the RPL protocol for supporting passive monitoring in the internet of things', in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, IEEE, pp.366–374.

Meng, W., Li, W., Xiang, Y. and Choo, K-K.R. (2017) 'A bayesian inference-based detection mechanism to defend medical smartphone networks against insider attacks', *Journal of Network and Computer Applications*, Vol. 78, pp.162–169.

Midi, D., Rullo, A., Mudgerikar, A. and Bertino, E. (2017) 'Kalis – a system for knowledge-driven adaptable intrusion detection for the internet of things', in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*,IEEE, pp.656–666.

Milenkoski, A., Viera, M. et al. (2015) 'Evaluating computer intrusion detection systems: a survey of common practices', *ACM Comput. Surv.*, September, Vol. 48, No. 1, Article 12, 41pp.

Miller, S., Curran, E. and Lunney, T. (2015) 'Traffic classification for the detection of anonymous web proxy routing', *International Journal for Information Security Research*, Vol. 5, No. 1, pp.538–545.

Moamed, T. and Helmi, B.M.R. (2016) 'Ant colony optimization and feature selection for intrusion detection', *Advances in Machine Learning and Signal Processing Power*, pp.305–312.

Natesan, P., Rajalaxmi, R.R., Gowrison, G. and Balasubramanie, P. (2017) 'Hadoop based parallel binary bat algorithm for network intrusion detection', *International Journal of Parallel Programming*, Vol. 45, No. 5, pp.1194–1213.

Omran, O.M.B. and Panda, P. (2016) 'Facilitating secure query processing on encrypted databases on the cloud', *IEEE International Conference on Smart Cloud (SmartCloud)*, pp.307–312.

Osanaiye, O., Cai, H., Choo, K-K.R., Dehghantanha, A., Xu, Z. and Dlodlo, M. (2016) 'Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing.', *EURASIP Journal on Wireless Communications and Networking*, Vol. 2016, No. 1, p.130.

Pajouh, H.H., Dastghaibyfard, G.H. and Hashemi, S. (2017) 'Two-tier network anomaly detection model: a machine learning approach', *Journal of Intelligent Information Systems*, Vol. 48, No. 1, pp.61–74.

Pajouh, H.H., Javidan, R., Khayami, R., Dehghantanha, A and Choo, K-K.R. (2016) 'A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks', *IEEE Transactions on Emerging Topics in Computing*.

Parihar, L.S. and Tiwari, A. (2016) 'Survey on intrusion detection using data mining methods', *International Journal for Science and Advance Research in Technology*.

Patel, R. and Aluvalu, R. (2014) 'A reduced error pruning technique for improving accuracy of decision tree learning', *International Journal of Engineering and Advanced Technology (IJEAT)*, ISSN, June, Vol. 3, No. 5, pp.2249–8958.

Pongle, P. and Chavan, G. (2015) 'Real time intrusion and wormhole attack detection in internet of things', *International Journal of Computer Applications*, Vol. 121, No. 9.

Ramos, A., Lazar, M., Filho, R.H. and Rodrigues, J.J.P.C. (2017) 'A security metric for the evaluation of collaborative intrusion detection systems in wireless sensor networks', in *2017 IEEE International Conference on Communications (ICC)*, IEEE, pp.1–6.

Sajid, A., Abbas, H. and Saleem, R. (2016) 'Cloud-assisted IoT-based scada systems security: a review of the state of the art and future challenges', *IEEE Access*, Vol. 4, pp.1375–1384.

Sforzin, A., Mármol, F.G., Conti, M. and Bohli, J-M. (2016) 'RPiDS: paspberry Pi IDS – a fruitful intrusion detection system for IoT', in *Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, *2016 Intl IEEE Conferences*, IEEE, pp.440–448.

Shah, V. and Agrawal, A.K. (2016) 'Enhancing performance of intrusions detection system against KDD99 datasets using evidence theory', I*nternational Journal of Cyber Security Digital Forensic*.

Sharma, N. and Gaur, B. (2016) 'An approach for efficient intrusion detection for KDD dataset: a survey', *International Journal of Advanced Technology and Engineering Exploration*, Vol. 3, No. 18, p.72.

Sherasiya, H.U. and Patel, H.B. (2016) 'A survey: intrusion detection system for internet of things', *International Journal of Computer Science and Engineering (IJCSE)*, Vol. 1, No. 5, pp.81–90.

Shidore, S.A. and Bhusari, V.K. (2014) 'Evasion of network intrusion detection system using functional framework', *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, Vol. 3, No. 6.

Sultana, A. and Jabbar, M.A. (2016) 'Intelligent network intrusion detection system using data mining techniques', in *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, pp.329–333.

Surendar, M. and Umamakeswari, A. (2016) 'InDReS: an intrusion detection and response system for internet of things with 6LoWPAN', in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, pp.1903–1908.

Thanigaivelan, N.K., Nigussie, E., Kanth, R.K., Virtanen, S. and Isoaho, J. (2016) 'Distributed internal anomaly detection system for Internet-of-Things', in *13th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, IEEE, pp.319–320.

Vamsi, P.R. and Kant, K. (2016) 'Trust aware data aggregation and intrusion detection system for wireless sensor networks', *International Journal on Smart Sensing and Intelligent Systems*, Vol. 9, No. 2, pp.537–562.

Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M. and Fischer, M. (2015) 'Taxonomy and survey of collaborative intrusion detection', *ACM Comput. Surv.*, Vol. 47, No. 55, pp.1–55.

Wang, K., Du, M., Sun, Y., Vinel, A. and Zhang, Y. (2016a) 'Attack detection and distributed forensics in machine-to-machine networks', *IEEE Network*, Vol. 30, No. 6, pp.49–55.

Wang, K., Du, M., Yang, D., Zhu, C., Shen, J. and Zhang, Y. (2016b) 'Game-theory-based active defense for intrusion detection in cyber-physical embedded systems', *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 16, No. 1, p.18.

Wang, Q., Yiğitler, H., Jäntti, R. and Huang, X. (2016c) 'Localizing multiple objects using radio tomographic imaging technology', *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 5, pp.3641–3656.

Wei, W., Yang, A.T. and Shi, W. (2016) 'Security in internet of things: opportunities and challenges', in *International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, IEEE, pp.512–518.

Wurzenberger, M., Skopik, F., Landauer, M., Greitbauer, P., Fiedler, R. and Kastner, W. (2017) 'Incremental clustering for semi-supervised anomaly detection applied on log data', in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ACM, p.31.

Yousif, D.M. and Hussein, M.A. (2014) 'Analyzing NB, DT and NBTree intrusion detection algorithms', *Journal of Zankoy Sulaimani-Part A (JZS-A)*, Vol. 16, No. 1.

Zarpelão, B.B., Miani, R.S., Kawakani, C.T. and de Alvarenga, S.C. (2017) 'A survey of intrusion detection in internet of things', *Journal of Network and Computer Applications*.

Zarpelão, B.B., Miani, R.S., Kawakani, C.T. and de Alvarenga, S.C. (2017) 'A survey of intrusion detection in internet of things', *Journal of Network and Computer Applications*, Vol. 84, pp.25–37.

Zuech, R., Khoshgoftaar, T.M. and Wald, R. (2015) 'Intrusion detection and big heterogeneous data: a survey', *Journal of Big Data*, Vol. 2, No. 3, pp.2–42, Springer.

View publication stats