ISSN - 2320-6349

International Journal of Mechanical Engineering & Computer Applications

# Artificial Intelligence Inspired Intrusion Detection System

Raheel Hussain[1], Maqsood Hayat[1]*, Nigar Fida[1], Mohammad Sohail[2]*, Muhammad Noman Hayat[1]
[1]Department of Computer Science, Abdul Wali Khan University, Mardan, KPK, Pakistan
[*2]Department of Physics, Abdul Wali Khan University, Mardan, KPK, Pakistan
Email: [1]m.hayat@awkum.edu.pk, [2]sohail.dagiwal@gmail.com

***Abstract*——** Intrusion Detection System (IDS) is a very important research area in the field of information and network security. In this paper, we discuss, analyze and evaluate some of the classifiers based on Artificial Intelligence (AI) which are used to detect and classify networks attacks for an intrusion detection system. We will use WEKA (A machine learning tool) software for testing and evaluating the performance of the classifiers on a well-known famous IDS (Intrusion Detection Systems) dataset "NSL-KDD" dataset. The NSL-KDD dataset contains sufficient amount of network attacks and it was developed in a military network by MIT Lincoln Labs. Once we got the individual results of all the classifiers then we will compare their performance on the basis of some performance parameters. We also used discretize filter to enhance the overall performance of the classifiers

***Index Terms*—** Intrusion Detection System, Weka, Detection Accuracy

## I. INTRODUCTION

With the advent of internet the world has become a global village where distances do not matters for communication. The usage of computer networks and internet are rapidly increasing for the purpose of information sharing and hence the technology is also evolved and improved to provide efficient data transfer, but on the other the security challenges are also increasing and more opportunities are made for cyber-crimes and attacks. These security threats compelled the users, organizations and government agencies to take some measures for protecting their systems and information from these threats, attacks and intrusions.

The ability to compromise the confidentiality, integrity, availability and quality of service of a system is called intrusion [1]. Different protection and defense mechanisms like authentication, firewalls, Antivirus and physical security are employed by organizations to prevent their systems and important data or information from attacks and intrusions. These protection mechanisms are good but they cannot provide protection against unknown complicated attacks, e.g. buffer overflow attacks chick makes use of the weakness in an application and cause massive security threat and intrusion. At this point the need for an Intrusion Detection System (IDS) appeared [2].

A system which has the capability to detect the un-authorized users who are trying to break the security of the network devices and systems is called Intrusion Detection System (IDS) [1].

Typically there are two types of IDSs used for computer network, that is Signature based and Anomaly based IDS [3], and some are hybrid as they are mix of those two [8]. A signature based IDS contains a signature database through it detects and identifies an attack, as each signature represents a known attack. Such types of IDS are as efficient and effective as their signature database. For detecting new threats and attacks the signature database of these IDSs must be updated with the required signatures. On the other hand anomaly based IDS learns the behavior of the system as normal or abnormal by using Artificial Intelligence (AI) techniques. In other words it builds the profile of the normal system and then detects the deviating or changed behavior of the system and identifies it as an attack or possible attack. These IDSs must be properly trained first by using machine learning techniques and should be upgraded on regular basis. The biggest advantage of anomaly based IDS is that it can detect the unknown non-trivial intrusions and attacks but there are more chances of generating false alarms [4]. So the challenges and problems in current anomaly based IDSs are efficiency and detection accuracy of intrusions [9]. In this work we are planning to study, analyze and evaluate the performance of some Artificial Intelligence (AI) based classifiers on the basis of efficiency and accuracy. We will also try to enhance and improve the efficiency and detection accuracy by using some filter mechanism. At the end we will compare their results with each other from different aspects.

This paper is organized as follows. Section 2 is the related works done on Intrusion detection System (IDS), section 3 is IDS approaches and techniques, section 4 is the discussion on a well-known dataset for IDS i.e. NSL-KDD Dataset, section 5 is about Weka software(a machine learning tool), section 6 is the study of selected classification algorithms (Classifiers) and techniques, section 7 is the performance evaluation of Artificial Intelligence (AI) based classifiers, section 8 is the results and discussion and section 9 is conclusion.

## II. RELATED WORK

For the very first time the concept of anomaly based intrusion detection system was introduced in [10], where the researchers developed a model that detects the behavior of the users and identifies it as a normal or anomaly or

intrusion. A framework that uses data mining techniques to detect anomalies and intrusions was presented in [5]. The concept of training multiple data mining classifiers and algorithms by using a set of attacks and malicious executables in order to detect new attacks and intrusions was presented in [11]. A study of all ruled based classifiers is carried out in [7] to evaluate their performance on the basis of detection accuracy, specificity, time, sensitivity and error.
A comprehensive study and analysis of KDD '99 dataset which is an intrusion detection system dataset is presented in [12]. The classification of NSL-KDD dataset by using Random Tree classifier and evaluation of its performance is done in [6]. Variety of techniques based on data mining i.e. Naive Bayes, CART and artificial neural network and their performance evaluation is presented in [13].

A comparative study of Naive Bayes, J48, OneR, PART, and RBF network classifier using NSL-KDD dataset is presented and the advantages of NSL-KDD dataset over KDDCUP'99 is also discussed in [17]. The study of intelligent classifiers i.e. Naive Bayes, Random Forest and Random Tree and their performance evaluation using NSL-KDD dataset is carried out in [2], they also made efforts to improve and enhance the detection accuracy of the classifiers by using some filter mechanisms.

## III. INTRODUCTION TO IDS

Intrusion Detection System has got a lot of popularity and importance in military, government and business organizations security and hence became an important and effective security system that monitors and analyses the data and information collected from different systems and internet resources which are connected with it and identifies them as an attack or a normal activity depending upon the nature of the data and information. This data compensate in the process of dealing with security attacks or intrusions [14]. This monitoring process is ubiquitous, because the intrusion detection system must be updated or changed as the security attacks change. A large number of security attacks and threats emerge from inside the organization, because of authorized users of the organization, mostly angry and offended employees. There is a possibility that the attacks happen with stolen credentials of an authorized employee, which can be very difficult to detect and trace. The outside users can also launch attacks like DOS (denial of service) attacks or hacking attacks to break into the organizations network or information system. Intrusion detection system is the only remedy to detect and encounter insiders as well as outsider's attacks [2].
Now a day's Intrusion detection systems (IDS) has become an essential part of the security infrastructure of the organizations. As the time passes the cyber-attacks increases. In 2013, hackers attacked the most famous social networking website Facebook. In early 2014 the logins information of 233 million users of eBay is hacked and in the same year the Yahoo e-mail service of 273 million users was hacked. In 2011, hackers attacked Sony and stole private and confidential information of more than a million users. In 2000, DOS (Denial of Service) attack was launched against Amazon and E-bay. According to Marianne Kolbasuk McGee, the executive editor of Information Security Media Group's HealthcareInfoSecurity.com media site, the biggest health data breach of 2015 is the cyber attack on health insurer Anthem Inc. - affected nearly 79 million individuals, making it, by far, the biggest healthcare breach on the list since its inception in late 2009. And the top six hacker attacks affected a combined total of 90 million individuals. A health insurer company Primera Blue Cross, based in Washington State, said that up to 11 million customers have become victims of a cyber attack in 2015. According to the New York Times, Cyber attacks have become an ever-increasing threat. The F.B.I. now ranks cybercrime as one of its top law enforcement activities, and the U.S President Obama's recently proposed that budget would sharply increase spending on cyber security, to $14 billion.
These consistent and recent fatal attacks show that an intrusion detection system is essential for the better security of information and data resources of organization, especially commercial networks and websites.

### A. TYPES OF IDS

IDS is proactive in nature, as it provides continuous ongoing monitoring of the system. IDS is application specific, meaning that there are special purposes or levels of security for which IDS has been designed. There are two types of IDS widely used now days.

I. HOST BASED IDS (HIDS)

Host-based intrusion detection systems (HIDS) are intended to collect information about activity on a particular single system, or host [14]. Host-based intrusion detection systems (HIDSs) take decisions on the basis of events collected by the hosts they monitor. The classification of HIDS depends on the type of audit data they analyse or on the techniques used to analyse their input.

II. NETWORK BASED IDS (NIDS)

Network-based intrusion detection systems (NIDSs) gather the input data by analysing and monitoring the network traffic or exchange of data between the computers in a network (e.g., packets captured by network interfaces in loose mode).

### B. IDS APPROACHES AND TECHNIQUES

For each of the two types of IDS- HIDS and NIDS, the most widely used basic techniques are Misuse (Signature) based detection and Anomaly based detection.

#### *I. MISUSE/ SIGNATURE BASED IDS*

Signature based IDS are pattern- based IDS that uses the exact match method for intrusion detection. This type of IDS can detect only those intrusions and attacks which are already known to it by their signatures.

*Advantage*

As signature based IDS follow the exact match criteria for the already known intrusion's detection, so the tendency of false alarm is very low. This type of IDS doesn't involve any type learning process or model building process so the performance will also be enough efficient and fast detection will be provided.

*Disadvantage*

Signature based IDS contains a signature database which must be updated with the signatures of new attacks and intrusions, otherwise the IDS will not be able to detect the

new attacks and intrusions. These signatures are developed by research teams e.g. Snort, Enterasys and Cisco etc.

*II. ANOMALY BASED IDS*

Anomaly based detection of intrusion is done by building or learning the profile of the normal and abnormal behaviours of the system. An anomaly-based intrusion detection system might build a baseline of how and when computers communicate across the network. All future communications will then be compared against the “normal” baseline of communications to determine if anomalous activity is occurring. If a user behaves in an unusual or suspicious way, the system will identify it as an intrusion. For example if any user logs in more than 25 times a day, or accessing e-mail or other services that he is not allowed to access, or log in at an inappropriate timings e.g. after office hours or on holiday etc. So in this case it will be considered as an unusual behaviour and the IDS will alert the system administrator about the suspicious behaviour of the user [2].

*Advantage*:

Because anomaly-based intrusion detection systems detect the misuse and intrusion on the basis of behaviour of the network and system, so the type intrusion does not need to be previously known. In other words we can say that anomaly based IDS has the ability to detect new unknown intrusions that signature based IDS may not detect.

*Disadvantage*:

Because behaviour on a system or a network can vary widely, anomaly-based systems have the tendency to report a lot of false alarms. The art of effectively identifying “normal” activity vs. truly abnormal is extremely challenging.

## IV. NSL-KDD

In 1998 MIT Lincoln Labs wanted to research into intrusion detection in their DARPA Intrusion Detection Evaluation Program. They generated a variety of intrusions which are simulated in a military network that became the 1999 KDD dataset for intrusion detection. This data was the result of raw TCP dump data of nice weeks for a simulated U.S. Air Force LAN with a variety of network attacks. The attacks can be categorized as follows:

(1) DoS – Denial of service
(2) U2R - Unauthorized access from a remote machine
(3) R2L - Unauthorized access to local super-user privileges
(4) Probe - Surveillance and other probing.

DoS assaults are planned in a manner that it devours the transmission capacity of the entire system and will look like ordinary activity. The client to root (U2R) assault happens on a nearby framework to raise the client rights to that of the super client. Remote to neighborhood (R2L) assaults are endeavors to login to a PC or gadget from outside or from remote range. Test assault is done over the system to accumulate the subtle elements and data of various gadgets on the system.

The KDD preparing dataset comprises of 494,019 records where 97,277 (19.69%) were named 'typical', 391,458 (79.24%) are named DoS assaults, 4,107 (0.83%) as Probe, 1,126 (0.23%) as R2L and 52 (0.01%) are delegated U2R assaults. Every record in the dataset has 41 properties portrayed distinctive components and a mark was appointed to each either as an "inconsistency" sort or as "ordinary" sort [15] [16]. As the dataset is as of now named so we don't have to do any progressions and changes to the dataset.

NSL-KDD is the new and overhauled adaptation of KDD '99 information set. It has understood a portion of the innate and inbuilt issues of the KDD'99 information set whose subtle elements are specified in [12]. In spite of the fact that, this new form of the KDD information set still experiences a portion of the issues talked about by McHugh [18] and may not be an impeccable illustrative of existing genuine systems, on account of the absence of open information sets for system based IDSs, yet we consider and trust that it can be connected as a powerful standard information set to help specialists think about and assess distinctive interruption discovery techniques. Moreover, the quantity of records in the NSL-KDD prepare and test sets are sensible. This favorable position makes it moderate to run the examinations on the finish set without the need to haphazardly choose a little parcel. Subsequently, assessment consequences of various research work will be predictable and practically identical [19].

Preferences of NSL-KDD over KDD '99 Data Set

The NSL-KDD information set has the accompanying preferences over the first KDD information set [19]:

There are no repetitive records in the prepare set, so therefore the classifiers won't be one-sided towards more successive records and will give the legitimized comes about.

The test sets don't contain any copy records; hence, the execution of the learning classifiers won't be one-sided by the techniques which have better location rates on the regular records.

The number of chose records from every trouble level gathering is conversely relative to the rate of records in the first KDD information set. Subsequently, the grouping rates of unmistakable machine learning strategies fluctuate in a more extensive territory, which makes it more productive to have a precise assessment of various learning procedures.

The number of records in the prepare and test sets is sensible and intelligent, which makes it moderate and cheap to play out the trials on the finish set without the need to arbitrarily choose a little divide or utilize a few strategies for information set lessening or selecting a few elements. Thus, assessment consequences of various research works will be steady and equivalent.

## V. WEKA 3.6.13: DATA MINING SOFTWARE

Weka is acclaimed toolkit for machine learning and data mining that was originally developed at the University of Waikato in New Zealand. This software is developed in Java programming language. It is a collection of machine learning algorithms or artificial intelligence (AI) classifiers for data mining tasks. The algorithms or classifiers can either be applied directly to a dataset or called from your own 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. This software is really a very useful tool to evaluate the performance of different classifiers using different parameters. In this work we are using WEKA for the evaluation of Naïve Bayes, J48, RBF Network and IBk classifiers using NSL-KDD dataset.

## VI. Classification Algorithms

In this section we will briefly explain the function if selected classifiers.

### A. Naïve bayes

Naïve bayes is a simple classifier that uses the probabilistic approach based on Bayes theorem (Baye's Rule) with strong independent (naïve) evidences and assumptions between the features of what is being binary classified (with two states i.e. yes or no). Suppose we have more than one evidence for building our Naive Bayes model, we could run into a problem of dependencies, i.e., some evidence may depend on one or more of other evidences. For example, the evidence "dark cloud" for the event of raining is directly dependent on the evidence of "high humidity" in the atmosphere. However, including dependencies into the model will make it very complicated. This is because one evidence may depend on many other evidences. To make our life easier, we make an assumption that all evidences are independent of each other (this is why we call the model "naive").

### B. J48

J48 is also a machine learning classifier in Weka. It is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. C4.5 is an algorithm that builds a decision tree based on a set of labelled input data. The decision trees generated and build by C4.5 algorithm can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. C4.5 builds decision trees from a set of training data using the concept of information entropy. The training data is a set $S = s_1, s_2, s_3,$ .....of already classified samples. Each sample $S_k = x_1, x_2, x_3,$ .....is a vector where $x_1$,$x_2$ and $x_3$.... represent attributes and features of the sample. The training data is augmented with a vector $C = c_1, c_2, c_3,$.... where $c_1, c_2, c_3,$.... represent the class to which each sample belongs [20].

### C. RBF Network

Radial basis function (RBF) networks are a type of feedforward network with a long history in machine learning. It is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. They are used in function approximation, time series prediction, and control. Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non linear RBF activation function and a linear output layer. A common strategy is to train the hidden layer of the network using k-means clustering and the output layer using supervised learning [23].

### D. IBk

In pattern recognition, the *k*-Nearest Neighbors algorithm or *k*-NN is used for pattern recognition [21]. It's a non-parametric method used for both classification and regression. The input of the k-NN is always consists of *k* neighboring training examples in the attribute space. The output of k-NN depends on whether it is used for classification or regression:

- In *classification*, the output of k-NN would be a class membership. An object is classified by a majority vote of its neighbours, and ultimately assigned to the most common class among its *k* nearest neighbours (*k* is a positive integer, typically small).
- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbours [22].

In Weka this algorithm is called IBk (Instance Based Learner). The IBk algorithm does not build a model; instead it generates a prediction for a test instance just-in-time. The IBk algorithm uses a distance measure to k- nearest neighbouring instances in the training data for each test instance and uses those selected instances to make a prediction.

*k*-NN is a type of instance-based or lazy learning mechanism, where the function is only approximated locally and all computation is delayed until classification is done. The *k*-NN algorithm is one of the simplest of all machine learning algorithms.

## VII. PERFORMANCE EVALUATION OF IDS CLASSIFIERS

In this section we will perform experiments and evaluate the performance of selected classifiers in weka by using a well known dataset NSL-KDD, which is widely used for intrusion detection system. Once we have done all the experiments and simulation then we will compare the resultsof the classifiers with each other on the basis of different performance parameters.

### A. Experiment Setup

All the experiments are conducted on Dell Laptop with Inter® Core I (3), 2.40 GHz Processor and 2.00 GB RAM, 450 GB HDD, windows 7 home basic with 64-bit operating system. In all the experiments, we used Weka 3.6.13to measure the accuracy, time taken to build models etc for Naïve Bayes, J48, RBF Network and IBk classifiers using NSL-KDD dataset. We have used "Cross Validation 10 folds" method for classification.

### B. Parameters for Performance Evaluation

The performance of the classifiers is evaluated on the basis of the following parameters or metrics:

- ✓ Accuracy
- ✓ Error Rate
- ✓ Model Building Time
- ✓ Average True Positive
- ✓ Average False Positive
- ✓ Average Recall
- ✓ Average F-Measure

### C. Discretization using Discretize Filter

After running the normal classifiers on the available datasets, we tried apply discretize filter on the dataset as a pre-processing before implementing the classifiers. In discretization the dataset id divided into a set of predefined intervals and groups the attribute values according to those interval values. In other words, the process of discretization is to divide the attribute's values of the dataset into a number of intervals so that each interval can be treated as single value of a discrete attribute [2]. And as a result the learning

complexity will be reduced drastically as evident from the results given in Table 1 and Table 2. The experiment was done as before.

**Results and Discussion**

We have used the selected classifiers with the two datasets i.e. 20% NSL-KDD Dataset and Full NSL-KDD Dataset in two ways. First the classification algorithms are experimented by using the datasets as it is they are already without any preprocessing. And then all the classifiers are implemented and run on the same datasets but this time with discretize filter as preprocessing of the datasets. The combined results of the experiments on both the datasets are shown intable1 and table2.

| Parameters | Naïve Bayes | Naive Bayes with Discretize filter | J48 | J48 with Discretize filter | RBF Network | RBF Network with Discretize filter | IBk | IBk with Discretize filter |
|---|---|---|---|---|---|---|---|---|
| Correctly Classified Instances | 22570 (89.59%) | 24323 (96.55%) | 25081 (99.56%) | 24997 (99.23%) | 23347 (92.68%) | 24416 (96.91%) | 25051 (99.44%) | 25104 (99.65%) |
| Incorrectly Classified Instances | 2622 (10.40%) | 869 (3.45%) | 111 (0.44%) | 195 (0.77%) | 1845 (7.32%) | 776 (3.08%) | 141 (0.56%) | 88 (0.35%) |
| Total Number of Instances | 25192 | 25192 | 25192 | 25192 | 25192 | 25192 | 25192 | 25192 |
| Root mean squared error | 0.3152 | 0.176 | 0.0651 | 0.0856 | 0.2484 | 0.162 | 0.0748 | 0.0538 |
| Model Building Time | 0.28 seconds | 0.02 seconds | 2.73 seconds | 0.53 seconds | 4.28 seconds | 3.11 seconds | 0.04 seconds | 0.01 seconds |
| TP Rate | 0.896 | 0.966 | 0.996 | 0.992 | 0.927 | 0.969 | 0.994 | 0.997 |
| FP Rate | 0.106 | 0.038 | 0.004 | 0.008 | 0.076 | 0.03 | 0.006 | 0.004 |
| Recall | 0.896 | 0.966 | 0.996 | 0.992 | 0.927 | 0.969 | 0.994 | 0.997 |
| F-Measure | 0.896 | 0.965 | 0.996 | 0.992 | 0.927 | 0.969 | 0.994 | 0.997 |

TABLE1: PERFORMANCE OF AI CLASSIFIERS USING 20% NSL-KDD DATASET

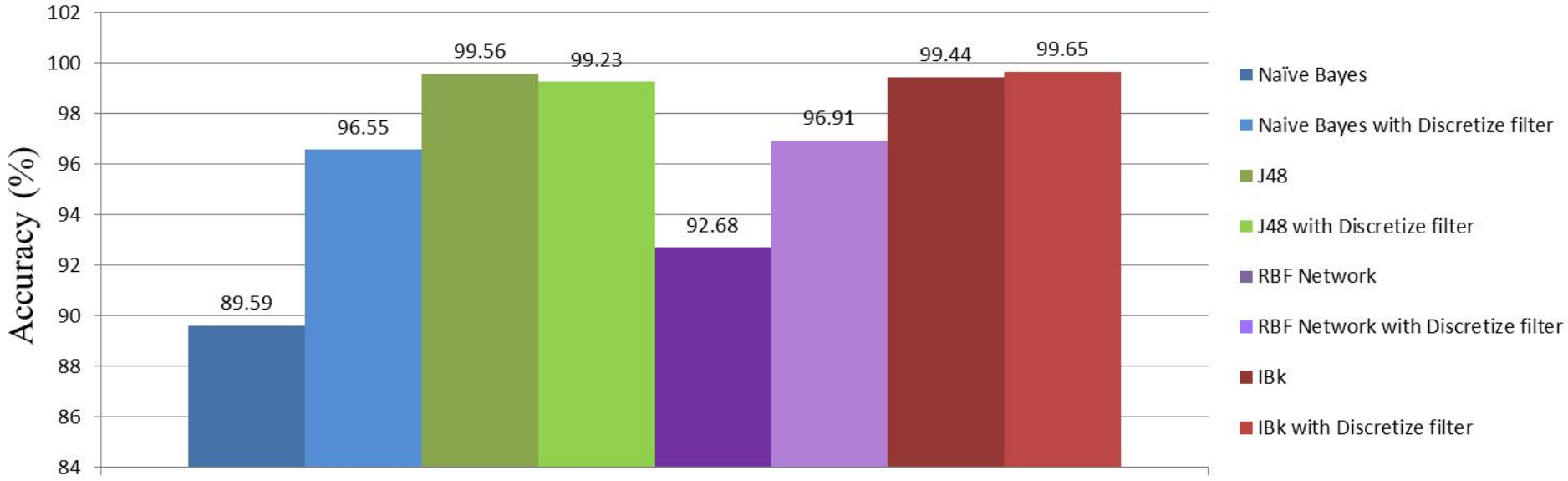

Figure 1: Accuracy of AI Classifiers using 20% NSL-KDD Dataset

| Parameters | Naïve Bayes | Naive Bayes with Discretize filter | J48 | J48 with Discretize filter | RBF Network | RBF Network with Discretize filter | IBk | IBk with Discretize filter |
|---|---|---|---|---|---|---|---|---|
| Correctly Classified Instances | 113858 (90.38%) | 122353 (97.13%) | 125698 (99.78%) | 125621 (99.72%) | 116753 (92.68%) | 122879 (97.54%) | 125652 (99.75%) | 125837 (99.89%) |
| Incorrectly Classified Instances | 12115 (9.62%) | 3620 (2.87%) | 275 (0.22%) | 352 (0.28%) | 9220 (7.32%) | 3094 (2.46%) | 321 (0.25%) | 136 (0.108%) |
| Total Number of Instances | 125973 | 125973 | 125973 | 125973 | 125973 | 125973 | 125973 | 125973 |
| Root mean squared error | 0.3058 | 0.1612 | 0.0457 | 0.0508 | 0.2474 | 0.1459 | 0.0504 | 0.0304 |
| Model Building Time | 1.11 seconds | 0.11 seconds | 34.51 seconds | 3.14 seconds | 30.66 seconds | 15.02 seconds | 0.08 seconds | 0.03 seconds |
| TP Rate | 0.904 | 0.971 | 0.998 | 0.997 | 0.927 | 0.975 | 0.997 | 0.999 |
| FP Rate | 0.101 | 0.032 | 0.002 | 0.003 | 0.076 | 0.024 | 0.003 | 0.001 |
| Recall | 0.904 | 0.971 | 0.998 | 0.997 | 0.927 | 0.975 | 0.997 | 0.999 |
| F-Measure | 0.904 | 0.971 | 0.998 | 0.997 | 0.927 | 0.975 | 0.997 | 0.999 |

TABLE2: PERFORMANCE OF AI CLASSIFIERS USING FULL NSL-KDD DATASET

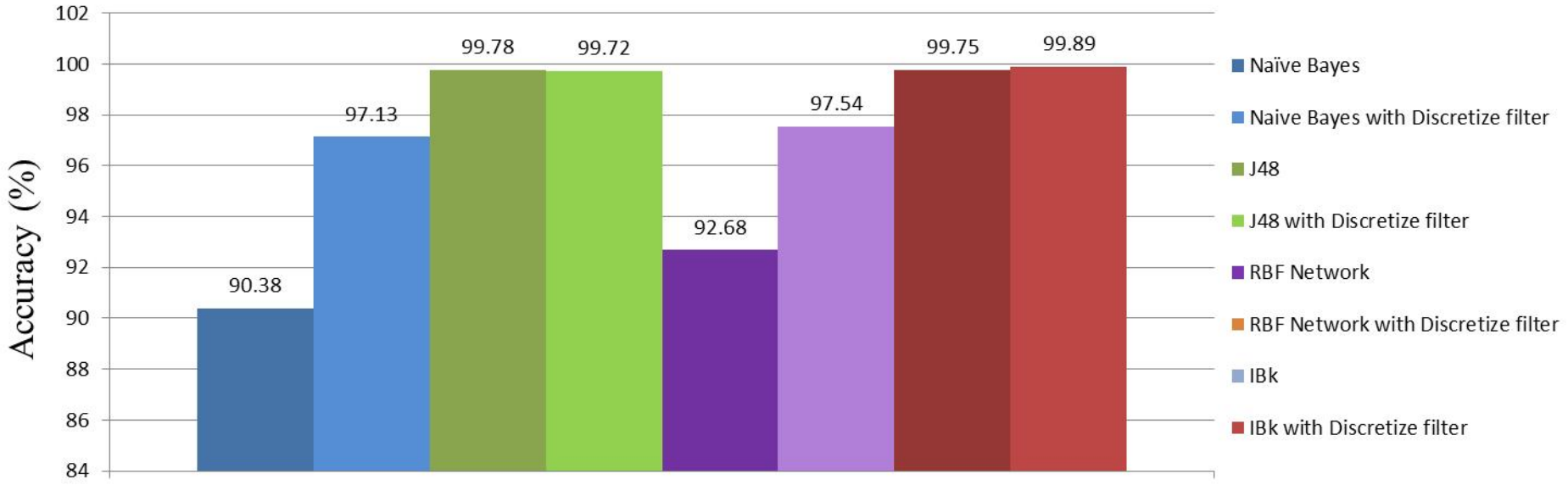


Figure 2: Accuracy of AI Classifiers using Full NSL-KDD Dataset

After compiling the results of the individual classifiers, we have observed the following observations for each individual classifier on the basis of performance metrics and parameters:

**A. Naïve Bayes**

- Naïve Bayes performs better with the large datasets comparatively because as the number of independent evidences increases the probability of occurrence of an event increases and hence accuracy. But on the other hand the time taken to build a model increases as the dataset becomes large as shown in table 1 and table 2.
- After applying the discritize filter to the dataset the performance of the naïve bayes classifier is very improved in terms of accuracy, True positive rate, False positive rate and model building time.

**B. J48**

- J48 classifier also performed better in case of large dataset but in this case it takes more time to build a model i.e. 34.51 seconds for normal full NSL-KDD dataset.
- After applying discritize filter this model building time is reduced significantly i.e. 3.14 seconds, but the accuracy is reduced very slightly, which is not a bad deal in return of efficient model building process as it is evident from table 1 and table 2.

**C. RBF Network**

- RBF Network classifier also gives satisfactory and almost same results in terms of accuracy for both small and large datasets, but the time required for building the model increases drastically when the dataset becomes large i.e. 30.66 seconds for Full NSL-KDD dataset, while the model building time for 20% NSL-KDD dataset is 4.28 seconds.
- After applying discritize filter to the datasets this model building time (i.e. 30.66 seconds) is reduced almost by half for full NSL-KDD dataset i.e. 15.02 seconds and 3.11 seconds for 20% NSL-KDD dataset as shown in table 1 and table 2. On the other hand the accuracy is also improved very well i.e. from 92.68% to 96.91% in case of 20% NSL-KDD dataset and from 92.68% to 97.54% in case of full NSL-KDD dataset.

**D. IBk**

- IBk (Instance Based) learning algorithm has given the best performance in terms of accuracy, root mean squared error, model building time, True positive rate and False positive rate and other performance parameters, for both small and large dataset or even more better in case of large dataset i.e. full NSL-KDD dataset.
- After applying the discritize filter the performance of the IBk classifiers is even more improved and enhanced. In case of 20% NSL-KDD the accuracy is improved from 99.44% to 99.65% while model building time is reduced from 0.04 seconds to 0.01 seconds, which is very good and efficient. In case of full NSL-KDD dataset the accuracy is improved from 99.75% to 99.89% while the model building time is reduced from 0.08 seconds to 0.03 seconds. This is the best performance among all the performances given by different classifiers.

## VIII. CONCLUSION

In this paper we have discussed the importance of information security and intrusion detection system (IDS) as an important and useful solution for the purpose of information security. We analyzed and evaluated different intrusion detection classifiers in weka by using NSL-KDD dataset. It has been observed that J48 which is a tress based classifier and IBk which is an instance based learning classifier using k- nearest neighboring method, performs better as far as accuracy is concerned. We also observed that J48 performs better for large datasets. After applying discretization on the dataset, the performance of the classifiers has improved highly in terms of both accuracy and model building time. IBk has given the best performance among all the classifiers we have evaluated in this paper. Although IBk does not build a model but still it can be used as an intrusion detection classifier. Furthermore, discretize filter is one of the best mechanism for improving the performance of the classifiers.

## REFERENCES


1. Panda, M., A. Abraham, and M.R. Patra, A hybrid intelligent approach for network intrusion detection. Procedia Engineering, 2012. 30: p. 1-9.
2. Albayati, M. and B. Issac, Analysis of Intelligent Classifiers and Enhancing the Detection Accuracy for Intrusion Detection System. International Journal of Computational Intelligence Systems, 2015. 8(5): p. 841-853.
3. Benferhat, S. and K. Tabia, Integrating Anomaly-Based Approach into Bayesian Network Classifiers, in e-Business and Telecommunications. 2009, Springer. p. 127-139.
4. Aydın, M.A., A.H. Zaim, and K.G. Ceylan, A hybrid intrusion detection system design for computer network security. Computers & Electrical Engineering, 2009. 35(3): p. 517-526.
5. Lee, W. and S.J. Stolfo. Data mining approaches for intrusion detection. inUsenix Security. 1998.
6. Subramanian, S., V.B. Srinivasan, and C. Ramasa, Study on classification algorithms for network intrusion systems. Journal of Communication and Computer, 2012. 9(11): p. 1242-1246.
7. G. V. Nadiammai and M. Hemalatha, (2012). "Perspective analysis of machine learning classifiers for detecting network intrusions," IEEE Third International Conference on Computing Communication & Networking Technologies (ICCCNT), India, pp. 1-7.
8. Tombini, E., et al. A serial combination of anomaly and misuse IDSes applied to HTTP traffic. in Computer Security Applications Conference, 2004. 20th Annual. 2004. IEEE.
9. Hofmann, A. and B. Sick, Online intrusion alert aggregation with generative data stream modeling. Dependable and Secure Computing, IEEE Transactions on, 2011. 8(2): p. 282-294.

10. Anderson, J.P., Computer security threat monitoring and surveillance. 1980, Technical report, James P. Anderson Company, Fort Washington, Pennsylvania.
11. Schultz, M.G., et al. Data mining methods for detection of new malicious executables. in Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on. 2001. IEEE.
12. Tavallaee, M., et al. A detailed analysis of the KDD CUP 99 data set. in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. 2009.
13. Srinivasulu, P., et al., Classifying the network intrusion attacks using data mining classification methods and their performance comparison. International Journal of Computer Science and Network Security, 2009. 9(6): p. 11-18.
14. Bace, Rebecca: An Introduction to Intrusion Detection & Assessment. Infidel Inc., prepared for ICSA Inc. Copyright 1998.
15. Siddiqui, M.K. and S. Naahid, Analysis of KDD CUP 99 dataset using Clustering based Data Mining. International Journal of Database Theory and Application, 2013. 6(5): p. 23-34.
16. KDD Cup 1999 Data (2014), Data and Task description, Online: http://kdd.ics.uci.edu/databases/kddcup99/ (accessed on May 2014).
17. Kalyani, G. and A.J. Lakshmi, Performance assessment of different classification techniques for intrusion detection. Learning, 2012. 2(1): p. J48.
18. McHugh, J., Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM transactions on Information and system Security, 2000. 3(4): p. 262-294.
19. NSL-KDD. (2014). The NSL-KDD Dataset. [Online] Available at: http://nsl.cs.unb.ca/NSL-KDD/ [Accessed: 13 Jan 2016]
20. Http://en.wikipedia.org/wiki/C4.5_algorithm
21. Altman, N.S., An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 1992. 46(3): p. 175-185.
22. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_ref-1
23. Http://en.wikipedia.org/wiki/Radial_Basis_function_network