Annals of Emerging Technologies in Computing (AETiC)
Vol. 2, No. 1, 2018

Article

A Comparative Study of Data Mining Algorithms for High Detection Rate in Intrusion Detection System

Nabeela Ashraf[1, *], Waqar Ahmad[2] and Rehan Ashraf[3]

[1]Department of Computer Science, National Textile University, Faisalabad, Pakistan
nabeela.ashraf@ymail.com
[2] Department of Computer Science, National Textile University, Faisalabad, Pakistan
waqarzahoor@gmail.com
[3] Department of Computer Science, National Textile University, Faisalabad, Pakistan
rehan_ashraf94@yahoo.com
*Correspondence: nabeela.ashraf@ymail.com

Received: 25th November 2017; Accepted: 15th December 2017; Published: 1st January 2018

Abstract: Due to the fast growth and tradition of the internet over the last decades, the network security problems are increasing vigorously. Humans can not handle the speed of processes and the huge amount of data required to handle network anomalies. Therefore, it needs substantial automation in both speed and accuracy. Intrusion Detection System is one of the approaches to recognize illegal access and rare attacks to secure networks. In this proposed paper, Naive Bayes, J48 and Random Forest classifiers are compared to compute the detection rate and accuracy of IDS. For experiments, the KDD_NSL dataset is used.

Keywords: Intrusion Detection System; Naive Bayes; J48; Random Forest; NSL_KDD dataset

1. Introduction

Today, internet plays a significant role in our professional and personal life. It is considered as the best foundation of information about the world. As some devices connected to the internet are increasing day-by-day, so internet security threats are also increasing. Having network access by using the internet, attackers have some ways to steal, destroy, or gain unauthorised data. The question is how to make our networks secure from such threats. Intrusion Detection System (IDS) is designed for this purpose [1, 2].

The basic purpose of IDS is to assist networks in coping with the network attacks so that they can identify anomalous, wrong and unsuitable actions of a computer system [3]. To protect systems from harmful attacks, network administrators use IDS. For security supervision, IDS became a crucial part. To hold operation normal throughout the harmful attack, intrusion detection system can identify and block harmful outbreaks [1].

Nabeela Ashraf, Waqar Ahmad and Rehan Ashraf, "A Comparative Study of Data Mining Algorithms for High Detection Rate in Intrusion Detection System", Annals of Emerging Technologies in Computing (AETiC), Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 49--57, Vol. 2, No. 1, 1st January 2018, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2018.01.005, Available: http://aetic.theiaer.org/archive/v2/v2n1/p5.html.

The most prominent problem of the traditional IDS is their incompetence to identify unique or unfamiliar attacks, as they are signature based [2]. Also, the problem that currently faces in Intrusion Detection System (IDS) is high false positives (FP) and high false negatives (FN). The intrusion detection system generates 15,000 number of alerts and thousands FP each day. Such mistakes that are generated by IDS lose the confidence of the network administrator and final user in security alerts [4].

One solution for the above problems is using data mining techniques that help to draw new patterns and procedures from the huge volume of audit data. Naive Bayes, J48 and Random Forest will be used to overcome these problems on the KDD_NSL dataset. The main objectives of this study are:

- Generate high accuracy rate on KDD_NSL dataset using Naive Bayes, J48, and Random Forest
- Generate high detection rate on KDD_NSL dataset using Naive Bayes, J48, and Random Forest

IDS is the act of identifying attacks that try to crack the confidentiality, integrity or availability (CIA) of a network. Intrusion Detection System is a software application or hardware device, collects and investigates different areas of network to identify the attacks and alerts the network administrator. It identifies both misuse attacks and intrusion attacks data. Intrusion attacks are those attacks that come from outdoor the institution, and misuse attacks come from inside the institution [2]. Two types of IDS are:

- Network-based (N-IDS)
- Host-based (H-IDS)

Network-based IDS are deployed at planned point/s inside the network to track and investigates the traffic from all appliances over the network. They are used to monitor the headers of transport and IP layer. In simple words, it protects the system from network-based attacks. Host-based IDS is run as a software application or agent that is deployed on individual hosts. The agents monitor the local operating system and alert the user or administrator if any suspicious activity is detected. H-IDS can only monitor the activities of the individual hosts on which the software application or agent is installed [2, 5]. In this review paper, we will pay attention to N-IDS approach.

Depend on network base detection; Intrusion Detection System can be

- Signature-based
- Anomaly-based

In Signature-based IDS, if any suspicious behaviour is found, it compares it in contrast to a database consisting of attributes or signatures from recognised malicious threats. It works like most antivirus software does. In intrusion detection system, the meaning of 'signature', is recorded proof of an attack or intrusion. It is also called knowledge-based IDS. The main issues with this IDS are it may be unsuccessful to detect new threats and needs to be updated as new signatures are recognised.

Anomaly-based IDS monitors the network packets, and if any suspicious behaviour is found in the packets, it compares them in contrast to a founded baseline. The baseline identifies the normality or abnormality of that network. If any suspicious activity or vulnerability is identified that deviates from the baseline, it alerts the administrator or user. It is also known as behaviour-based IDS. It is more responsive to new threats than the signature-based IDS but difficult to implement [2, 5].

Now the question is how IDS reacts to the detected threat? IDS have two reaction modes: passive or reactive. Passive IDS just monitor and analyse the network traffic activities, and if detect any threat or vulnerability it alerts the administrator or user to take a decision. Now it depends upon the administrator either to chunk the activity or react in some other fashion. Passive IDS cannot perform anything (protective or corrective) on its own. Reactive Intrusion Detection System is also recognised as Intrusion Detection and Prevention System (I-DPS). It not only monitors the network traffic or alerts the administrator about the suspicious activity but also responds to the threat by using pre-defined proactive actions. For instance; it blocks additional network traffic from the user or source IP address [2].

This paper is structured in this fashion: Section II describes the data mining algorithms used in this paper along with their related work in the context of IDS. Materials and Methods are presented in section III, where we discuss the dataset, preprocessing techniques, performance measure and implementation

flow of the proposed IDS scheme. Section IV gives the analysis of results performed on NSL_KDD dataset, and section V gives the conclusion and future work.

2. Literature Review

In this section, we focus on the classification techniques used in this study and also their related work that has done in IDS.

A. Naive Bayes

Naive Bayes is a supervised learning classifier that based on Bayes' Theorem with the "naive"' notion of independence among variables of a problem. This notion means that the presence of one variable in a problem does not have any effect on the presence of another variable. Naive Bayes uses the conditional probability. It classifies the problem by combining previous calculated likelihood and probabilities to make the next probability using Bayes rule [2]. Naïve Bayes contains two mechanisms. The first module is Directed Acyclic Graph (DAG). In DAG nodes called random variables represent the probabilistic dependencies. The second module is a set of parameter which defines the restricted likelihood. The restricted dependency represents DAG. It is used in text classification, spam detection, recommendation system etc..

Panda et al. [6] suggested an IDS constructed on Naive Bayes, accomplished improve results than IDS based backpropagation NN when verified on the KDD'99 dataset. Amor et al. [7] attained 91.52% overall accuracy by using Naive Bayes when verified on the KDD'99 dataset. Although its structure is simple, it can deliver accurate results. Farid et al. [8] suggested Naive Bayes and decision tree based hybrid intrusion detection system and accomplished detection rate of 99.63% on the KDD'99 dataset.

Dickerson et al. [9] designed FIRE Intrusion Detection System. To procedure network stimulus data, simple data mining techniques used and reveal important anomaly detection metrics. For every observed value, these metrics are accessed and used to identify network attacks after ahead. Bajaj et al. [10] achieved detection accuracy of 76.56% and 81.05% when applied Naive Bayes and J48 on NSL_KDD dataset to detect intrusions.

B. J48 and Random Forest

J48 is labelled as C4.5 algorithms. The gain ratio is used to produce a DT, and it is a non-binary tree. In J48 classifier, dataset spilt through the value of root node and value of root node depends upon the features having the highest value. Every node calculates its gain value separately, and this process of calculation carried until the process of prediction is carried. J48 is a version of C4.5 algorithms, and it is a decision tree classifier of Weka [11]. For attributes assessment, DT is produced originally based on feature values achieved through training data. The classification of data examples can be made based on attributes influence by trained test data [12]. Random Forest is a supervised learning classifier which based on a group of three analysts. Each tree produces a random selection and in training set is made from the examples of the classification tree. Every tree provided a classification and called "votes" for that class. Forest chooses the overall classification trees in the forest, by the elementary value that a group of "weak learner"' can compose from a "strong learner" [2].

Bouzida et al. [13] compared decision tree with PCA and without PCA. The principal component analysis is a numerical process that translates some correlated features into some uncorrelated features. With little damage to overall accuracy from 92.60% to 92.05%, by a factor of about thirty, they cut down computational time on the KDD'99 dataset. They determined that neural networks are good for generalisation but bad for detecting new attacks while decision trees are efficient in both generalisations and detecting novel assaults.

Ali et al. [14] conducted experiments on 20 multipurpose datasets to compare the results of random forest and J48 classifiers. Results showed that random forest gives better results for a large number of instances as compared to J48. They also discussed the effect of missing values difficulties in the datasets. By utilising random forest Classifier, Farnaaz et al. [15] developed a model for IDS and conducted experiments on the NSL_KDD dataset. To evaluate the results of their proposed model, they compared it with the J48 classifier. Results proved that random forest achieved better accuracy than J48.

Nutan et al. [1] enlisted 49 associated studies published between 2009 and 2014 concentrating on the application of utilising different classifiers for IDS. He concluded that better results could be achieved by the elimination of duplicate and inappropriate features from the datasets. Hybrid or ensemble classifiers should be used in performance measure instead of single or baseline classifiers.

3. Materials and Methods

A. Dataset

The commonly used dataset for IDS till now is a KDD'99 dataset. NSL_KDD is the enhanced description of KDD'99 datasets. We cannot say that NSL_KDD dataset is an ideal model of networks because it still experiences some problems examined by McHugh. However, still, researchers believe that it is the impressive model that can be used to detect intrusions on different models [6].

There are 39 different attacks, 42 attributes out of which 41 attributes are equivalent as in KDD'99, and the 42nd attribute is class tag [4], total 1, 48,516 records in NSL_KDD test dataset. In this paper, 20% of the dataset is used that contains 25,192 instances of the total dataset. Training dataset contains 15, 115 instances and testing dataset contains 6, 298 instances.

Table 1. Attack Classification in NSL_KDD Dataset

Sr. No.	Class	No. of Samples	Attack Types
1	**Denial of Services**	53385	teardrop, pod, neptune, apache2, processtable, smurf, land, udpstorm, back, mailbomb
2	**User to Root Attack**	252	perl, ps, rootkit, xterm, loadmodule, buffer_overflow, sqlattack, httptunnel
3	**Remote to User Attack**	3749	multihop, warezclient, named, ftp_write, snmpguess, sendmail, warezmaster, imap, worm, xsnoop, guess_passwd, xlock, phf, spy, snmpgetattack
4	**Probe**	14077	nmap,saint, portsweep, satan, mscan, ipsweep
5	**Normal**	77053	normal

Different kinds of assaults in NSL_KDD dataset lie into following four classes:

- Denial of services attacks (DOS):

This is the type of attacks where intruder makes some memory or computing resources unavailable so that eligible users can't access them. Intruders can introduce DoS attacks in different ways like by corrupting the computer's certain features, by attempting to execute viruses, or by abusing the system's structure. For DoS attacks classification see Table 1.

- User to Root Attacks (U2R):

In this type of attacks, the intruder begins by getting access to the regular user account on the system and achieve root access to the system by exploiting the susceptibility. For U2R attacks classification see Table 1.

- Remote to User Attacks (R2L):

This type of attacks occurs when intruder transmits packets to a machine through the network, then makes use of system's susceptibility to getting local access to the system as a user. R2L attacks are classified as shown in Table 1.

- Probe:

The probe is the class of attacks where intruder inspects networks to acquire evidence or discover familiar susceptibilities. This type of inspection is very helpful for the intruder who is executing an attack in the future. An intruder who has a report of machines and facilities that are available on a specified network can avail this evidence to look for feeble ideas [3, 11]. See its attacks classification on Table 1.

The percentage of different instances in each attack class and normal class is shown in Figure 1.

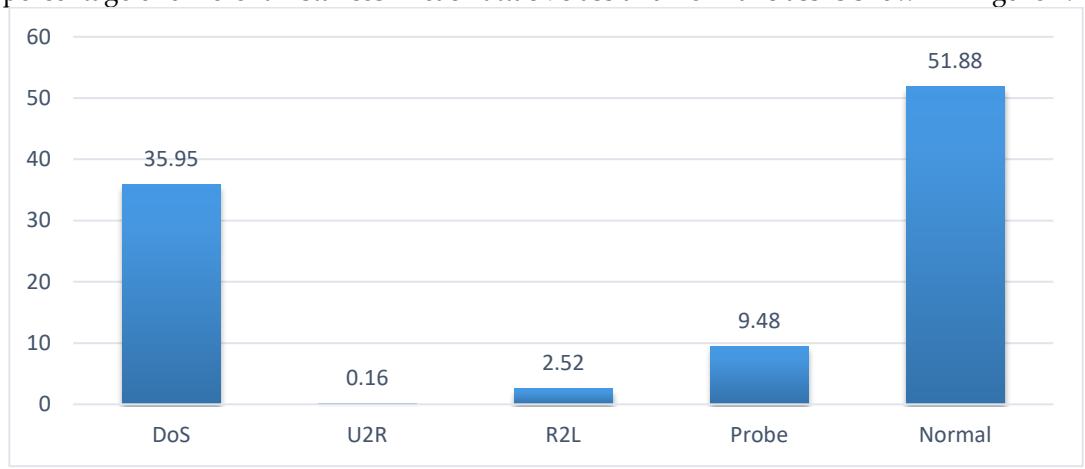

Figure 1. Number of Instances in Each Class

B. Preprocessing

It is a very important process in data mining. It is used to remove useless attributes from the dataset. This process is performed in two phases. In the first phase, valuable attributes are selected. There are two main methods to select attributes: wrapper and Filter. We used filter method where Info Gain is used as an attribute evaluator and ranked as a search method in Weka. After applying this filter, 23 attributes are left from 42 attributes.

Discretization step has been proved to be beneficial for many classifying algorithms that deal with only nominal values. By using this, continuous feature values are grouped into a predefined set of intervals. So in the second step, I applied discretization process on the dataset. The effect of discretization process on Count Variable is shown in Figure 2 and 3.

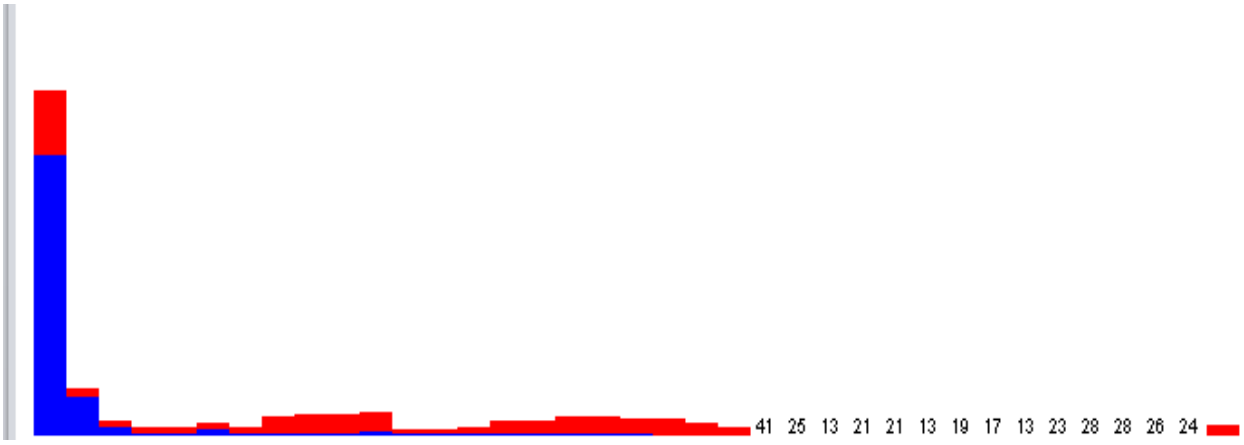

Figure 2. Count Variable (Before Discretization)

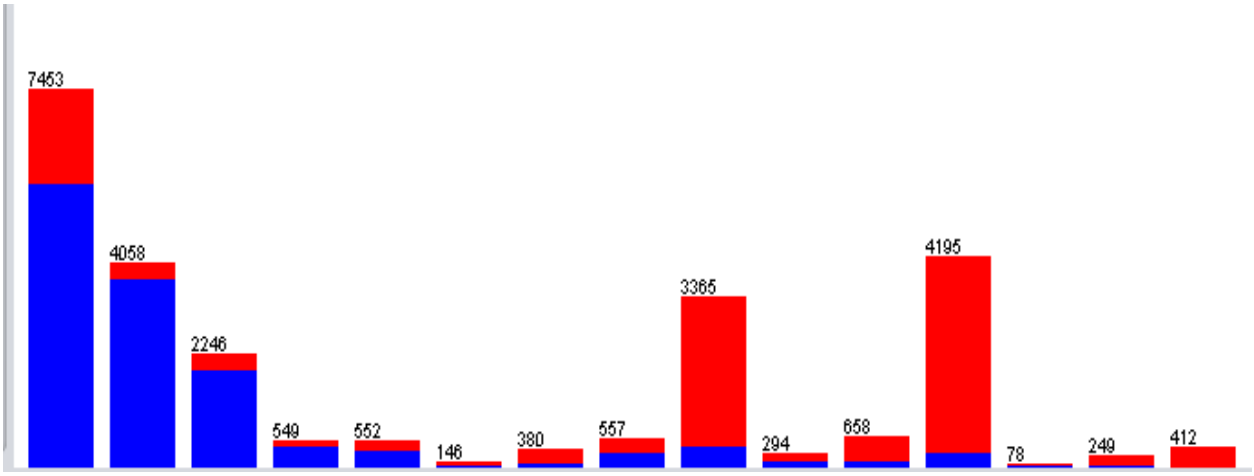

Figure 3. Count Variable (After Discretization)

C. Performance Measure

To measure the performance of each classifier following metrics are required:

- Detection Rate (DR): DR (Recall) represents the percentage of intrusions that are correctly detected. It is calculated by:

$$DR = (TP / (TP + FN)) * 100$$

- Overall Accuracy (OA): OA (Precision) represents the percentage that is designed by the total number of intrusions that are properly determined allocated by the total amount of considerations [16]:

$$OA = ((TP+TN)/(TP+TN+FP+FN))*100$$

Where:

TP: is True Positive that occurs when an occasion is correctly determined as intrusion

TN: is True Negative that occurs when an occasion is correctly determined as normal

FN: is False Negative that arises when an event is considered as normal, but actually, it is intrusive

FP: is False Positive that arises when an event is considered as intrusive, but actually, it is normal.

An effective intrusion detection has high Recall and Precision with low false measures.

- F-Measure: It is also called F-score. It is used to evaluate the correctness of a test. It is a mixture of Recall and Precision. Best results are achieved when F-measure equal to 1 and worst when F-measure is 0 [16]. It is given as:

  F-Measure = (Precision (OA) * Recall (DR)) / (Precision (OA) + Recall (DR))

D. Implementation Flow

The proposed method is described below:

1. Load 20% NSL_KDD dataset on Weka 3.8
2. To extract valuable attributes from the dataset, apply filter method as described in subsection B. After filtration process, 23 attributes are left
3. Now apply discretization process to obtain continuous feature values
4. Divide the dataset into training and test data
5. Train the dataset by applying classifier algorithm (Naive Bayes/ Random Forest/ J48)
6. For classification, test data is fed to trained dataset
7. Measure Precision, Recall and F-Measure.

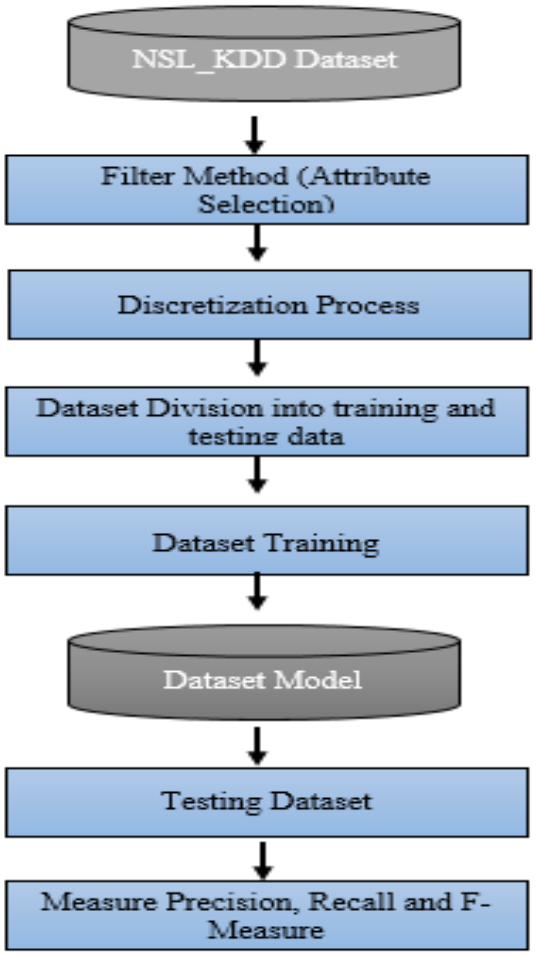

Figure 4. Implementation flow Diagram

4. Results and Analysis

In this paper, the performances of Naive Bayes, J48 and Random Forest tree are compared. 10-fold cross-validation is used to evade overfitting. Table 2 presents the correctly classified and incorrectly classified instances for the Naive Bayes, J48 and Random Forest tree classifiers. The results depict that Random Forest achieved better results than Naive Bayes and J48 on the same dataset.

Table 2. Performance of Naive Bayes, J48 and Random Forest Classifiers on 20% NSL_KDD Dataset

	Correctly Classified Instances	**Incorrectly Classified Instances**
Naive Bayes	96.27%	3.73%
J48	99.17%	0.83%
Random Forest	99.71%	0.29%

Table 3 presents the overall accuracy (Precision), detection rate (Recall) and F-Measure for the Naive Bayes, J48 and Random Forest. From this table, it can be seen clearly that Random Forest tree performs better than Naive Bayes and J48. Random Forest has achieved maximum F-Score, i.e. 99.7%.

Table 3. Comparison of Naive Bayes, J48 and Random Forest regarding Precision, Recall and F-Measure

	Precision	**Recall**	**F-Measure**
Naive Bayes	0.964	0.963	0.963
J48	0.992	0.992	0.992
Random Forest	0.997	0.997	0.997

5. Conclusion

Because of the fast growth of the internet, many techniques are used to avoid intrusions. However, still, there is a need to develop more efficient systems that detect new or unique intrusions. In this paper, the classification performances of Naive Bayes, J48 and Random Forest are compared on 20% KDD_NSL dataset. From the results, we conclude that Random Forest performed better than Naive Bayes and J48 regarding both accuracy and detection rate. All the three classifiers achieved up to 90% results in both precision and recall so, in future, we can combine these three techniques and compute the result. We can further incorporate the outcome of this research work to improve the border security network as well [17].

References

[1] N. F. Haq, A. R. Onik, A. K. Hridoy, M. Rafni, F. M. Shah, and D. Farid, "Application of Machine Learning Approaches in Intrusion Detection System: A Survey," *Int. J. Adv. Res. Artif. Intell.*, vol. 4, no. 3, pp. 9–18, 2015.

[2] M. Stampar, "Artificial Intelligence in Network Intrusion Detection." *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015

[3] H. Tribak, B. L. Delgado-Marquez, P. Rojas, O. Valenzuela, H. Pomares, and I. Rojas, "Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System," *2012 Int. Conf.*

Multimed. Comput. Syst., pp. 434–440, 2012.
[4] M. Tausif, J. Ferzund, S. Jabbar, R. Shahzadi, "Towards Designing Efficient Lightweight Ciphers for Internet of Things", *KSII Transactions on Internet and Information Systems*, Vol. 11, No. 8, 2017
[5] A. Carlin, M. Hammoudeh, and O. Aldabbas, "Intrusion Detection and Countermeasure of Virtual Cloud Systems -State of the Art and Current Challenges," *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 6, 2015.
[6] M. Panda and M. R. Patra, "Network Intrusion Detection Using Naïve Bayes," *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 258–263, 2007.
[7] N. Ben Amor, S. Benferhat, and Z. Elouedi, "Naive Bayesian Networks in Intrusion Detection Systems."
[8] D. M. Farid, N. Harbi, and M. Z. Rahman, "Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection," May 2010.
[9] J. E. Dickerson and J. A. Dickerson, "Fuzzy network profiling for intrusion detection," in *PeachFuzz 2000. 19th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.00TH8500)*, pp. 301–306.
[10] K. Bajaj and A. Arora, "Dimension Reduction in Intrusion Detection Features Using Discriminative Machine Learning Approach.," … *J. Comput. Sci. Issues (IJCSI* …, vol. 10, no. 4, pp. 324–329, 2013.
[11] A. Osareh and B. Shadgar, "Intrusion Detection in Computer Networks based on Machine Learning Algorithms," *Ijcsns*, vol. 8, no. 11, p. 15, 2008.
[12] H. P. S. Sasan and M. Sharma, "Intrusion Detection Using Feature Selection and Machine Learning Algorithm with Misuse Detection," *Int. J. Comput. Sci. Inf. Technol.*, vol. 8, no. 1, pp. 17–25, 2016.
[13] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, and S. Gombault, "Efficient Intrusion Detection Using Principal Component Analysis."
[14] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 5, pp. 272–278, 2012.
[15] N. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," *Procedia Comput. Sci.*, vol. 89, pp. 213–217, 2016.
[16] M. E. Elhamahmy, H. N. Elmahdy, and I. A. Saroit, "A New Approach for Evaluating Intrusion Detection System," *Int. J. Artif. Intell. Syst. Mach. Learn.*, vol. 2, no. 11, pp. 290–298, 2010.
[17] M. Hammoudeh. 2016. Putting the lab on the map: A wireless sensor network system for border security and surveillance. In *Proceedings of the International Conference on Internet of things and Cloud Computing (ICC '16)*. ACM, New York, NY, USA.