Proc. of the Intl. Conf. on Future Trends in Computing and Communication -- FTCC 2013
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-981-07-7021-1

# *Accuracy Analysis of Machine Learning Algorithms for Intrusion Detection System using NSL-KDD Dataset*

Sanoop Mallissery, Sucheta Kolekar, Raghavendra Ganiga
Department of Information & Communication Technology
Manipal Institute of Technology
Manipal University, Karnataka, India

**Abstract—Intrusion Detection System (IDS) that turns to be a vital component to secure the network. The lack of regular updation, less capability to detect unknown attacks, high non adaptable false alarm rate, more consumption of network resources etc., makes IDS to compromise. This paper aims to classify the NSL-KDD dataset with respect to their metric data by using the best six data mining classification algorithms like J48, ID3, CART, Bayes Net, Naïve Bayes and SVM to find which algorithm will be able to offer more testing accuracy. NSL-KDD dataset has solved some of the inherent limitations of the available KDD'99 dataset.**

**Keywords- IDS, KDD, Classification Algorithms, PCA etc.**

## I. INTRODUCTION

In the future Internet age, computer security has become the core foundation for most of the applications like online marketing, online transfers, etc. Intrusion detection is the technique used to detect the attacks on computer or network by examining various data's observed in the network traffic. It is one of the important ways to solve network security problems. Correctly classify the instances and detection precisions are the two basic measures to evaluate intrusion detection systems (IDS) [1]. In order to enhance the classification of instances and detection precision, many works have been done. Most of the earlier research was focusing on rule-based expert systems and statistical approaches. But when working with larger datasets, the results of rule-based expert systems and statistical approaches become worse. This will be pointing to the data mining algorithms to solve the problems [2]. A common problem of Network IDS (NIDS) is that it detects only the known services or network attacks only, which is called misuse detection, by using pattern matching approaches. But on the other side an anomaly detection system detects attacks by making profiles of normal networks or system behaviors first, and then identifies the attacks if the behaviors are significantly deviated from the normal system or network profiles. Many methods have been proposed in the past few years for the design of effective NIDSs, among which, decision trees have been proven to give a good performance.

The main idea behind the usage of data mining methods in intrusion detection is the automation. Data mining techniques, such as decision trees, naïve Bayesian classifiers, neural networks, support vector machine, k-nearest neighbors, fuzzy logic model, and genetic algorithm have been widely used to analyze network logs to gain intrusion related knowledge to improve the performance of IDS in last decades [3]. To apply data mining techniques in intrusion detection, first the collected network logs or audit data needs to be preprocessed and converted to the format that suitable for mining. Next, the reformatted data will be used to develop a clustering or classification model. Data mining provide decision support for intrusion management, and also help IDS for detecting new vulnerabilities and intrusions by discovering unknown patterns of attacks or intrusions. Principal component analysis (PCA) is an essential technique in data compression and feature selection [4] which has been applied to the field of Intrusion Detection [5, 6]. PCA is an efficient method to reduce dimensionality by providing a linear map of n dimensional feature space to a reduced m-dimensional feature space [7]. In this paper, PCA is applied for feature dimension reduction.

## II. RELATED WORKS

Many researchers have applied data mining techniques for the efficient design of NIDS. Data mining applications involve millions or even billions of pieces of data records. For example, in the KDD Cup'99 dataset, there are more than 4 million and 3 million instances in the training set and test set, respectively. But some of the techniques are not able to apply on such larger datasets due to the insufficient memory capacity of the system or time taken to finish the training. The clustering method could produce high quality dataset with far less instances that sufficiently represent all of the instances in the original dataset. Here we had used the NSL-KDD dataset. NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD Cup'99 data set which are mentioned in [8, 9]. It is very difficult to signify existing original networks, but still it can be applied as an effective benchmark data set for researchers to compare different intrusion detection methods [10].

To overcome the weakness of signature based IDSs in detecting the novel attacks, researchers has attracted to anomaly detection. KDD cup'99 dataset is most widely used for the evolution of these systems. In [9] they have conducted a statistical analysis on this data set and found two important issues which highly affect the performance of evaluated system, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, they proposed a new dataset, NSL-KDD, which consists of selected records of the complete KDD dataset and does not suffer from any of the mentioned shortcomings.

Network security is becoming an increasingly important issue, since the rapid Development of internet. As the main security defending technique Network Intrusion Detection Systems (NIDS) is widely used against such malicious attacks [11]. Data mining and machine learning technology has been extensively applied in network intrusion detection and prevention system by discovering user behaviour patterns from the network traffic data. Association rules and sequence rules are the main techniques of data mining for intrusion detection.

Selecting the relevant set of attributes for data classification is one of the most significant problems in designing a reliable classifier. Existing C4.5 decision tree technology has a problem in their learning phase to detect automatic relevant attribute selection, while some statistical classification algorithms require the feature subset to be selected in a pre-processing phase. Also, C4.5 algorithm needs strong pre-processing algorithm for numerical attributes in order to improve classifier accuracy in terms of Mean root square error. In [12] they have evaluated the influence of attribute pre-selection using statistical techniques on real-world KDD cup'99 data set. Experimental result shows that accuracy of the C4.5 classifier could be improved with the robust pre-selection approach when compare to traditional feature selection techniques.

Irrespective of whether good anomaly detection methods are used, the problems such as high false alarm rates, difficulty in finding proper features, and high performance requirements still exist. Therefore, if we are able to mix the advantages of both learning schemes in machine learning methods, according to their characteristics in the problem domain, then the combined approach can be used as an efficient means for detecting anomalous attacks.

## III. Classification Algorithms

### A. *Support Vector Machine (SVM)*

SVM performs classification by constructing an N-dimensional hyperplane that separates the data into two categories optimally. SVM is very closely related to the concepts of neural networks. In fact the sigmoid kernel function of SVM is equivalent to a two-layer perceptron neural network [13].

SVM consists of two types of attributes, a predictor variable, also called as an attribute and a transformed attribute using to define a hyperplane known as feature. Choosing the most suitable representation from this is called as feature selection. The set of features used to describe a row of predictor values is called as a vector. The goal of SVM modeling is to get an optimal hyperplane that separates clusters of vector. The separation will be like, one category of the target variable is on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the *support vectors*. Figure 1 presents the SVM process overview.

SVM is insisting on finding the maximum margin hyperplanes and that provides to offer good generalization ability. It also provides a very accurate classification performance over the training records and generates enough search space for the accurate classification of future data. For generating an optimal margin hyperplane, SVM classifier will maximize equation (1) with respect to vector $V$ and constant term $\mu$. Where $Lpg$ is the Lagrangian with $\alpha_j$ Lagrange multipliers which uses $t$ number of training samples from $j = 1,2,\ldots t$ and the vectors $V$ and $\mu$ will represent the hyperplane.

$$Lpg = \sum_{j=1}^{t} \alpha_j + \frac{1}{2}\|V\| - \sum_{j=1}^{t} \alpha_j y_j \left(V \cdot x_j + \mu\right) \qquad (1)$$

SVMs are always reasonable to the appropriate selection of parameters. So it always ensures a series of parameter combinations no less than on a sensible subset of the data. In SVM it's better to scale the data always; because it will drastically improve the results. So be careful with large dataset, because it may leads to the increase in training time.

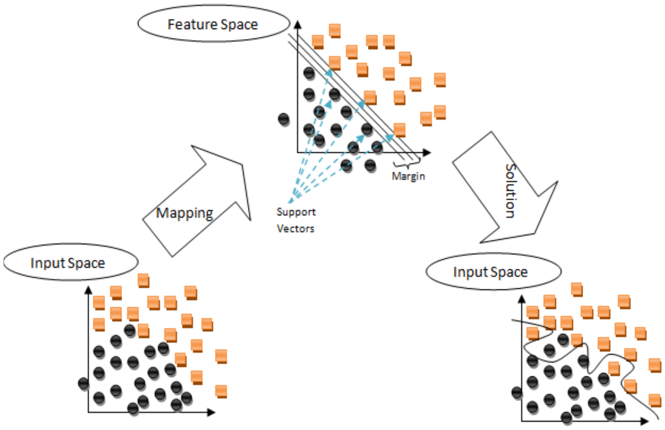

Figure 1. Overview of SVM process

### B. *J48*

J48 algorithm is developed for the WEKA and MONK project [16]. The algorithm is an extension for C4.5 decision tree algorithm. There are many options for tree pruning in case of J48 algorithm. The classification algorithms available in WEKA try to simplify the results or prune. This method will help us to produces more generic results and also can be used to correct potential overfitting issues. J48 helps to recursively classify until each of the leaf is getting pruned, that is to categorize as close knit to the data. So this will helps to ensure the accuracy, but excessive rules will be generated. But pruning will cause to less accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.

J48 employs two pruning methods. The first is known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf -- basically reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second type of pruning used in J48 is termed subtree raising. In this case, a node may be moved upwards towards the root of the tree, replacing other nodes along the way. Subtree raising often has a negligible effect on decision tree models. There is often no clear way to predict the utility of the option, though it may be advisable to try turning it off if the induction process is taking a long time. This is due to the fact that subtree raising can be somewhat computationally complex.

Error rates are used to make actual decisions about which parts of the tree to replace or raise. There are multiple ways to do this. The simplest is to reserve a portion of the training data to test on the decision tree. The reserved portion can then be used as test data for the decision tree, helping to overcome potential overfitting. This approach is known as reduced-error pruning. Though the method is straight-forward, it also reduces the overall amount of data available for training the model. For particularly small datasets, it may be advisable to avoid using reduced error pruning.

### *C. Iterative Dichotomiser 3(ID3)*

ID3 is a non-incremental algorithm, meaning it derives its classes from a fixed set of training instances. An incremental algorithm revises the current concept definition, if necessary, with a new sample. The classes created by ID3 are inductive, that is, given a small set of training instances, the specific classes created by ID3 are expected to work for all future instances. The distribution of the unknowns must be the same as the test cases. Induction classes cannot be proven to work in every case since they may classify an infinite number of instances [15].

A statistical property, called information gain, is used. Gain measures how well a given attribute separates training examples into targeted classes. The one with the highest information (information being the most useful for classification) is selected. In order to define gain, we first borrow an idea from information theory called entropy Suppose it's given a collection $H$ on class $K$ then we will get an equation as follows in (2), where $P(K)$ is the proportion of $H$ belonging to class $K$.

$$Entropy(H) = H - p(K)\log_2 P(K) \qquad (2)$$

### *D. Classification and Regression Trees (CART)*

The concept of CART is very similar to C4.5 algorithm in data mining. The main difference between CART and C 4.5 is that it's supporting the concept of regression and it is not computing the rule sets [13]. CART also uses the idea of binary trees by using threshold and as well as the feature that yield to produce the largest information gain at each and every node.

The idea of tree growing in CART is based on the decision to split among all the probable splits at each and every node and this will results to a purest child. In CART only univariate splits will be considering then each split will depends on one predictor variable value. Suppose $Y$ is a nominal categorical variable of $j$ then the possible number of splits will be $2^{j-1}-1$ for this particular predictor. Supposing that $Y$ is a continuous variable or an ordinal group with $X$ different values then $X-1$ different splits will be there on $Y$. The decision tree will start growing constantly from the root node by using the following steps on each node.

1. Determine every predictor's best split.
2. Determine the node's best split.

From the finest splits found in step 1, select the splitting criterion that will show the maximum.

3. If at all the stopping rules are not fulfilled, then split the nodes by using its best possible split found in step 2.

At node $v$, the best split $q$ will be choosing to maximize a splitting criterion $\Delta i(q,v)$. The impurity measure for a particular node helps to determine the splitting criterion that corresponds to a decrease in the impurity measure revealed. Where $\Delta I(q,v) = p(v)\Delta i(q,v)$ is referred to as an improvement, with probability $p(v)$ of a case in node $v$.

### *E. Bayes Net*

Bayes nets [16] are networks of relationships and its shows the basic law of probability which is now called Bayes rule as in (3). For example consider any two events named X and Y, then the probability becomes as shown in equation (3).

$$P(Y \mid X) = P(X \mid Y) * P(Y) / P(X) \qquad (3)$$

Bayes net relates nodes which are probabilistically points to causal dependency and it will end up with a huge saving of computation time. Because of its adaptability Bayesian nets are proved that it is so useful. In a Bayes net the links may form loops but it will not form any cycles. But it is not an expressive drawback that won't be bounding the modeling power, but should be careful while building nets. This will leads to a major advantage that it will very fastly updating the algorithms since if there is no further way to control this in a probabilistic manner to cycle entirely for an indefinite period.

### *F. Naïve Bayes*

Naive Bayes' model is a conditional independence model in which each predictor gives the target class more accurately [13]. The Bayesian principle is used to predict a situation of a class that shows the most important posterior probability. Bayesian approach helps to decide the document class $r$ as the only one that will maximize the conditional probability $P(Cn \mid r)$ in equation (4).

$$P(Cn \mid r) = \frac{P(r \mid Cn)P(Cn)}{P(r)} \qquad (4)$$

For calculating $P(r \mid Cn)$, we have to make a Naïve Bayes assumption and all the attributes should not be statistically dependent. Here $r$ can act as a vector of $m$ number of attribute values and it will leads to an assumption that shown in equation (5).

$$P(r \mid Cn) = \prod_{i=1}^{m} P(r_i \mid Cn) \qquad (5)$$

Calculate $P(r_i \mid Cn)$ for $i$ values with a proportion of documents from class $Cn$ that comprise attribute values $r_i$. Probability of sampling $P(Cn)$ that included in class $Cn$ will be calculated as a proportion of all data's in the training documents. If only the class label has to be determined then the general denominator $P(r)$ is not required in the calculations. Consider there are $k$ terms ( $v_1, v_2, ..., v_k$ ) and $m$ documents ( $do_{1,} do_2, ..., do_m$ ) which corresponds to each and every attributes in the description given in the document respectively from class $Cn$. Where $m_{ij}$ denotes that how many times the term $v_i$ occurs in a particular document $do_j$ . $P(v_i \mid Cn)$ is the probability that the description term $v_i$ occurs in the documents from class $Cn$. Hence we can calculate how many times the term $v_i$ occurs in all the respective document from the class $Cn$ is shown in equation (6).

$$P(v_i \mid Cn) = \frac{\sum_{j=1}^{m} m_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{m} m_{ij}} \qquad (6)$$

## IV. Statistical Observations

The statistical analysis of KDD data set shows some important issues that degrade the evaluation of anomaly detection approaches and it will affect the performance of the evaluated system. In KDD dataset there are 3663472 intrusions and 159967 normal repeated records in the training set. The test set consists of 221058 intrusion and 12680 repeated records in normal. This leads replacement of KDD data set to NSL-KDD data set. In NSL-KDD dataset they have removed all the redundant data's present in the KDD'99 dataset. So NSL-KDD dataset consists only very selected records. But this selected data's in NSL-KDD dataset is much sufficient to provide a good analysis compared to KDD Cup'99 dataset. All the analysis process observations are by using the WEKA software [17]. The data analysis and attack classification was carried out using WEKA software environment for machine learning. WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules and visualization. It is also well-suited for developing new machine learning schemes.

The data in NSL-KDD dataset is either labeled as normal or as one of the 24 different kinds of attack. These 24 attacks can be grouped into four classes: Probe, DoS, R2L, and U2R. The following steps are carried out in WEKA:

i. Select the dataset
ii. Run the six classifier algorithms
iii. Compare the six classifiers.

The first step is to perform discretization. Discretization is the process of turning numeric attributes into nominal attributes by converting the numeric values into a small number of distinct ranges. An instance filter discretizes a range of numeric attributes in the dataset into nominal attributes. The main benefit is that some classifiers can only take nominal attributes as input, not numeric attributes. Another advantage is that some classifiers that can take numeric attributes can achieve improved accuracy if the data is discretized prior to learning. The next step is without using filters; perform the process for the above classifiers. Table I shows the correctly and incorrectly classified instances by using the above six algorithms.

TABLE I. Classified Instances

| Classification Algorithm | Classified Instances (%) | |
|---|---|---|
| J48 Algorithm | Correctly Classified | 89.77 |
| | Incorrectly Classified | 10.23 |
| ID3 Algorithm | Correctly Classified | 92.23 |
| | Incorrectly Classified | 7.766 |
| SimpleCART Algorithm | Correctly Classified | 88.73 |
| | Incorrectly Classified | 11.26 |
| BayesNet Algorithm | Correctly Classified | 67.05 |
| | Incorrectly Classified | 32.95 |
| NaïveBayes Algorithm | Correctly Classified | 67.53 |
| | Incorrectly Classified | 32.47 |
| SVM Algorithm | Correctly Classified | 99.80 |
| | Incorrectly Classified | 0.2 |

There are three kinds of symbolic features (tcp, ftp_data and SF etc.) in feature space of 41 features. We are not giving much importance for these three feature vectors in our work and are discarded to get the following new feature space to 38. From this we have filtered to 23 feature vectors by using PCA technique to get an optimum selection from complete dataset with 41 features for training as well as for testing experiments. Table II shows the test accuracy that achieved by using the six algorithms for the full dimension data and also after the feature reduction with PCA technique This shows that PCA can be used with any classification algorithms without much reduction in the test accuracy.

TABLE II. Test Accuracy for Different Classes of Attacks

| Classification Algorithms | Class Names | Test Accuracy (%)with 41 Features | Test Accuracy (%)with 23 Features |
|---|---|---|---|
| SVM | Normal | 99.1 | 99.8 |
| | DOS | 98.8 | 99.5 |
| | U2R | 90.6 | 81.6 |
| | R2L | 93.4 | 73.1 |
| | Probe | 94.1 | 97.6 |
| ID3 | Normal | 92.3 | 94.8 |
| | DOS | 93.1 | 96.3 |
| | U2R | 82.1 | 73.2 |

| | | | |
|---|---|---|---|
| | R2L | 80.7 | 54.1 |
| | Probe | 87.1 | 70.7 |
| J48 | Normal | 73.1 | 77.7 |
| | DOS | 82.4 | 70.1 |
| | U2R | 69.7 | 50.6 |
| | R2L | 73.1 | 67.4 |
| | Probe | 80.2 | 69.3 |
| SimpleCART | Normal | 88.9 | 91.1 |
| | DOS | 82.7 | 80.8 |
| | U2R | 73.1 | 70.3 |
| | R2L | 80.8 | 63.4 |
| | Probe | 82.1 | 75.4 |
| BayesNet | Normal | 69.1 | 67.5 |
| | DOS | 68.4 | 63.4 |
| | U2R | 63.1 | 64.3 |
| | R2L | 72.1 | 69.2 |
| | Probe | 69.2 | 72.3 |
| NaïveBayes | Normal | 70.1 | 69.1 |
| | DOS | 72.7 | 59.2 |
| | U2R | 69.1 | 54.3 |
| | R2L | 68.5 | 62.1 |
| | Probe | 70.4 | 67.1 |

Figure 2 shows the test accuracy on class Normal attack that compared with 41 features and with the reduced set of features by using PCA. Here the SVM algorithm shows the highest accuracy compared with rest of the algorithms by considering with and without feature reduction. Figure 3 shows the test accuracy on class DOS attacks. As can be seen in Figure 4 and Figure 5, the test accuracy of SVM with reduced feature set is low compared to the test accuracy that has shown without feature reduction in the case of U2R & R2L attack classes. But compared to other algorithms SVM has shown much better performance to produce good test accuracy without much reduction. Figure 6 shows the test accuracy for probe attack class and SVM is able to produce better test accuracy in case of reduced feature set also.

## V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have used the NSL-KDD dataset that solves some of the snags of KDD99 dataset. Our analysis shows that NSL-KDD dataset is very ideal for comparing different intrusion detection models. Using all the 41 features in the network to evaluate the intrusive patterns may leads to time consuming detection and also the performance degradation of the system. Some of the features in this are redundant and irrelevant for the process. We have used the PCA technique for reduce the dimensionality of the data. Our experiment has been carried out with six different classification algorithms for the dataset with and without feature reduction and in that SVM shows a high test accuracy compared to all other algorithms in both the cases. So in the case of reduced feature set this analysis shows that SVM is speeding up the training and the testing methods for intrusion detection that is very essential for the network application with a high speed and even providing utmost testing accuracy. In future we can try to improve the SVM algorithm to build an efficient intrusion detection system.

## REFERENCES

[1] H. Debar, M. Dacier and A. Wespi, "Towards a taxonomy of intrusion-detection systems", Computer Networks, vol. 31, pp. 805-822, 1999.

[2] D. Marchette, "A statistical method for profiling network traffic". In proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring , pp. 119-128, 1999.

[3] S. Mukkamala, G. Janoski and A.Sung, "Intrusion detection: support vector machines and neural networks" In the Proc. of the IEEE International Joint Conference on Neural Networks (ANNIE), pp. 1702-1707, 2002.

[4] E. Oja, "Principal components, minor components, and linear neural networks", Neural Networks, vol. 5, pp. 927-935, 1992.

[5] G.K. Kuchimanchi, V.V. Phoha, K.S. Balagami and S.R. Gaddam, "Dimension reduction using feature extraction methods for Real-time misuse detection systems" In the Proc. of the IEEE Workshop on Information Assurance and Security, pp. 195-202, 2004.

[6] M. Shyu, S. Chen, K. Sarinnapakorn and L. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier" In the Proc. of ICDM'03, pp. 172-179, 2003.

[7] K. Labib and V.R. Vemuri, "Detecting and visualizing denial of-service and network probe attacks using principal component analysis" In the Proc. of the 3rd Conference on Security Architectures, 2004.

[8] "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSL-KDD.html, March 2009.

[9] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", In the Proc. of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009), pp. 1-6, 2009.

[10] 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, vol. 3, pp. 262–294, 1998.

[11] Lei Li, De-Zhang Yang, Fang-Cheng Shen, "A Novel Rule-based Intrusion detection System Using Data Mining", In the Proc. Of 3rd IEEE International Conference on Computer Sceince and Information Technology, pp. 169-172, 2010.

[12] K.Nageswara Rao, D.Rajya Lakshmi, T.Venkateswara Rao" Robust Statistical Outlier based Feature Selection Technique for Network Intrusion Detection" International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

[13] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, " Top Ten Data Mining Algorithms", Knowledge and Information Systems Journal, Springer-Verlag London, vol. 14, Issue 1, pp. 1-37, 2007.

[14] N.S. Chandolikar, Nandavadekar, "Efficient Algorithm for Intrusion Attack Classification by Analysing KDD Cup 99", In the Proc. of the 9th International Conference on Wireless and Optical Commuincations Networks (WOCN), pp. 1-5, 2012.

[15] Guangqun Zhai, Chunyan Liu, "Research and Improvement on ID3 Algorithm in Intrusion Detction System", In the Proc. of the 6th International Conference on Natural Computation (ICNC), vol. 6, pp. 3217-3220, 2010.

[16] Cooper, G.F., Herskovits, E.: A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In: Seventh Conference on Uncertainty in Artificial Intelligence, pp. 86–94, 1991.

[17] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009.

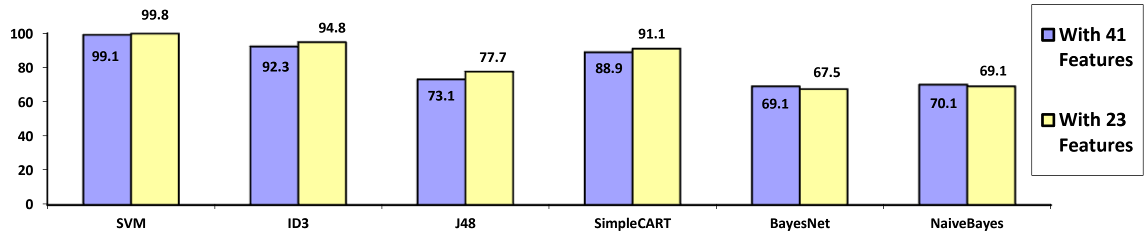

Figure 2. Test Accuracy of Class NORMAL Attack Comparing with 41 & 23 Features

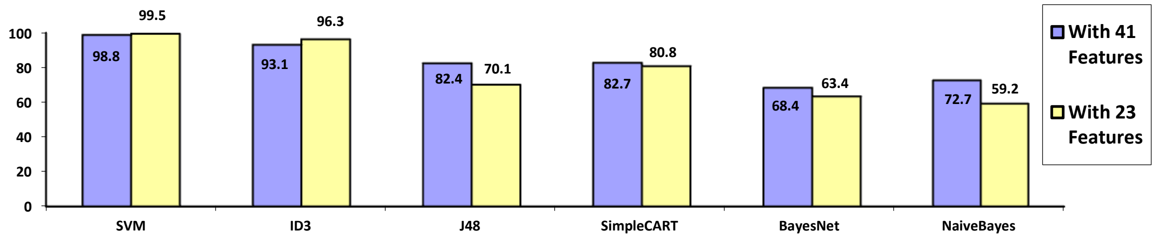

Figure 3. Test Accuracy of Class DOS Attack Comparing with 41 & 23 Features

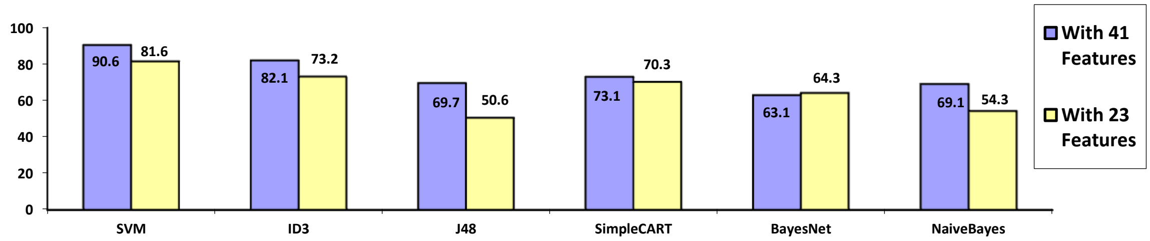

Figure 4. Test Accuracy of Class U2R Attack Comparing with 41 & 23 Features

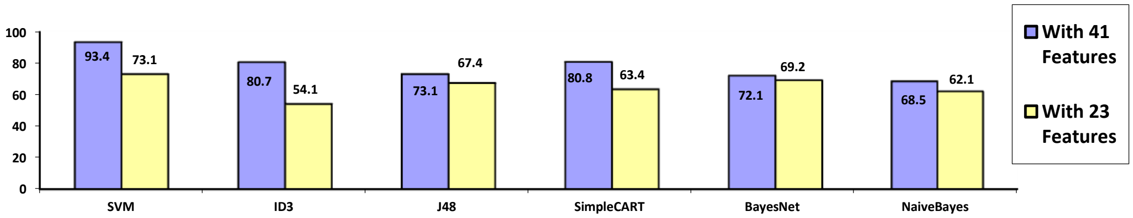

Figure 5. Test Accuracy of Class R2L Attack Comparing with 41 & 23 Features

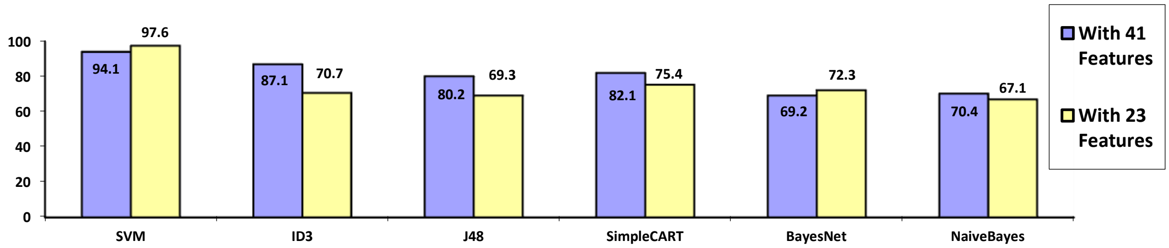

Figure 6. Test Accuracy of Class PROBE Attack Comparing with 41 & 23 Features