

Feature Selection for Modeling Intrusion Detection

Virendra Barot

I.T. Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad -388306, Gujarat, India
Email: viren.rao82@gmail.com

Sameer Singh Chauhan

I.T. Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad -388306, Gujarat, India
Email: onlyforsameer@gmail.com

Bhavesh Patel

I.T. Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad -388306, Gujarat, India
Email: bhavesh.svit@gmail.com

Abstract—Feature selection is always beneficial to the field like Intrusion Detection, where vast amount of features extracted from network traffic needs to be analysed. All features extracted are not informative and some of them are redundant also. We investigated the performance of three feature selection algorithms Chi-square, Information Gain based and Correlation based with Naive Bayes (NB) and Decision Table Majority Classifier. Empirical results show that significant feature selection can help to design an IDS that is lightweight, efficient and effective for real world detection systems.

Index Terms—Feature selection, network intrusion detection system, decision table majority, naive Bayesian classification.

I. INTRODUCTION

With the wide and quick development of network technology, in the field of social networking, e-business, e-learning and online shopping, Security is a big issue for all networks in today's enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over Internet. Some of them are the use of firewalls, encryption, and virtual private networks. Intrusion detection is a relatively new addition to such techniques. Intrusion detection methods with machine intelligence started appearing in the last few years. Using intrusion detection methods, you can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts.

An Intrusion Detection System (IDS) is the device (or application) that monitors network/system activities and the analyzing of data for potential vulnerabilities and attacks in progress; it also raises alarm or produces report [1]. Different sources of information and events based on information are gathered to decide whether intrusion has

taken place. This information is gathered at various levels like system, host, application, etc [2]. Based on analysis of this data, we can detect the intrusion based on two common practices – Misuse detection and Anomaly detection.

Misuse detection IDS models function in very much the same sense as high-end computer anti-virus applications. That is, misuse detection IDS models analyze the system or network environment and compare the activity against signatures (or patterns) of known intrusive computer and network behavior [3].

Anomaly detection takes the normal observation model and uses statistical variance [4] or expert systems to determine if the system or network environment behavior is running normally or abnormally.

The paper is organized as follows. Section 2 in our paper gives brief idea of the work done in this field i.e. intrusion detection using data mining and feature selection for it. In Section 3, we give brief detail of probability based Naive Bayesian model, Decision Table and various attribute selection scheme used in the experiments. In Section 4, we discuss the dataset used and experiment results in detail. Finally, we concluded the whole work in Section 5.

II. RELATED WORK

IDS have become important and widely used for ensuring network security. Since the amount of audit data that an IDS needs to examine is very large even for a small network, analysis is difficult even with computer assistance because extraneous features can make it harder to detect suspicious behavior patterns [5][9].

Data mining approaches can be used to extract features and compute detection model from the vast amount of audit data. The features computed from the data can be more objective than the ones handpicked by experts. The inductively learned detection model can be more generalized than hand-coded rules (that is they can have better performance against new variants of known normal

behavior or intrusions). Therefore data mining approaches can play an important role in process of developing Intrusion Detection Systems. Complex relationships exist between the features and IDS must therefore reduce the amount of data to be processed. This is very important if real-time detection is desired. Reduction can occur by data filtering, data clustering and feature selection. In complex classification domains, features may contain false correlations, which hinder the process of detecting intrusions. Extra features can increase computation time, and can have an impact on the accuracy of IDS.

Feature selection improves classification by searching for the subset of features, which best classifies the training data. In the literature a number of work could be cited wherein several machine learning paradigms, fuzzy inference systems and expert systems, were used to develop IDS [5][6].

Reference [7] has stated that Naïve Bayes classifiers provide a very competitive result even this classifier having a simple structure on his experimental study. According to the author, Naïve Bayes are more efficient in classification task.

Authors of [8] have demonstrated that large number of features is unimportant and may be eliminated, without significantly lowering the performance of the IDS. Hongjie Liu, Boqin feng, jianjie weng [10] in 2008, given the model that combines k means and decision table classifier. Authors of [11] have used correlation based feature selection and employed it in rule base intrusion detection based on support vector machine and decision tree.

III. ATTRIBUTE SELECTION AND CLASSIFICATION TECHNIQUES

Intrusion detection can be thought of as a classification problem: we wish to classify each audit record into one of a discrete set of possible categories, normal or a particular kind of intrusions. Given a set of records, where one of the features is the class label (concept), classification algorithms can compute a model that uses the most distinguishing (unique) feature values to describe each concept. So, classification tasks typically require the construction a function (classifier) that assigns a class label to each data item described by a set of attributes.

The term data mining is frequently used to designate the process of extracting useful information from large databases. There are a wide variety of data mining algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and databases. Several types of algorithms are particularly relevant to intrusion detection.

A. Bayesian Model

Naive Bayes classifier is the one among many variations of the Bayesian models available. It is a simple

probabilistic classifier, based on applying Bayes' theorem with strong (naive) independence assumptions.

Let D be a training set of tuples. Each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, A_1, A_2, \dots, A_n . Assume that there are m classes, C_1, C_2, \dots, C_m . Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that the tuple X belongs to the class C_i if and only if

$$P(C_i | X), i \in [1, m] > P(C_j | X),$$

$$\text{for all } 1 \leq j \leq m, j \neq i$$

Where, $P(C_i | X)$ is the probability of instance to fall in class C_i , given feature set value $[X_1, X_2, \dots, X_n]$.

The problem is that if the number of features n is large and when a feature can take on a large number of possible values too, then basing such a model on probability tables is infeasible.

Therefore reformulate the model to make it more tractable that lies on Bayes Theorem Equation

$$P(C_i | X) = \frac{P(C_i)P(X | C_i)}{P(X)} \quad (1)$$

Where, $P(X)$ = the constant for all classes so can be ignored.

$P(C_i)$ = the prior probability for class C_i .

To evaluate $P(X | C_i)$, the naïve assumption of class conditional independence is used. That is, each feature X_i is conditionally independent of every other feature X_j for $j \neq i$. It implies

$$P(X_i | C, X_j) = P(X_i | C)$$

$$\begin{aligned} \text{Thus, } P(X | C_i) &= P(X_1 | C_i)P(X_2 | C_i)..P(X_n | C_i) \\ &= \prod_{k=1}^{k=n} P(X_k | C_i) \end{aligned} \quad (2)$$

We can easily estimate the probabilities $P(x_1/C_i), P(x_2/C_i), \dots, P(x_n/C_i)$ from the training tuples available in the following two ways. If A_k is an categorical attribute, then

$$\begin{aligned} P(X_k | C_i) &= \frac{\text{No. of samples of class } C_i \text{ having value } X_k \text{ for } A_k}{\text{No. of total samples belongs to class } C_i} \end{aligned} \quad (3)$$

If A_k is an continuous-valued attribute, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute [4, 7] is typically assumed to have a Gaussian distribution with mean and standard deviation

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4)$$

B. Decision Table

A decision table is an organizational or programming tool for the representation of discrete functions. It can be viewed as a matrix where the upper rows specify sets of conditions and the lower ones sets of actions to be taken when the corresponding conditions are satisfied; thus each column, called a rule, describes a procedure of the type “if conditions, then actions”.

Given an unlabelled instance, decision table classifier searches for exact matches in the decision table using only the features in the schema (it is to be noted that there may be many matching instances in the table). If no instances are found, the majority class of the decision table is returned; otherwise, the majority class of all matching instances is returned.

If the training dataset size is, say D and test data set size is, say d with N attributes, The complexity of predicting one instance will be O (D*N). So, the underlying data structure used for bringing down the complexity is Universal Hash table. The time to compute the hash function is O (n') where n' is the number of features used as schema in decision table. So complexity will become lookup operation for n' attribute multiplied by l, number of classes that is O (n' + 1).

To build a decision table, the induction algorithm must decide which features to include in the schema and which instances to store in the body. More details can be found in [11]. We use CFS algorithm as induction algorithm for our experiment.

C. Correlation Based Feature Selection

This algorithm is a heuristic for evaluating the merit of a subset of features. The hypothesis on which the heuristic is based can be stated:

“Good feature subsets contain features highly correlated with the class, yet uncorrelated with each other.”

$$Merit_s = \frac{k \overline{r_{cf}}}{\sqrt{k + k(k-1) \overline{r_{ff}}}} \quad (5)$$

Where $Merit_s$ is the “merit” of a feature subset S containing k features, r_{cf} is the feature-class correlation (f e s), and r_{ff} is the feature-feature inter correlation.

Various search strategies such as hill climbing and best first are often applied to search the feature subset space in reasonable time. CFS starts from the empty set of features and uses a forward best first search with a stopping criterion of five consecutive fully expanded non-improving subsets. More details of feature correlation computation can be found in [15].

D. Chi-square based Attribute Selection.

Chi-square ([13, 14]) test is commonly used method, which evaluates features individually by measuring chi-square statistic with respect to the classes. The statistic is

$$\chi^2 = \sum_{i=1}^k \sum_{j=1}^n \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad (6)$$

Where, k = No. of attributes,

n = No. of classes,

A_{ij} = number of instances with value i for attribute and j for the class,

E_{ij} = the expected No. of instances for A_{ij} .

The larger value of the χ^2 , indicates highly predictive to the class.

E. Information Gain based Attribute Selection.

If X and Y are discrete random variables, (7) and (8) give the entropy [15, 16] of Y before and after observing variable X.

$$H(Y) = -\sum_i p(y_i) \log_2 p(y_i) \quad (7)$$

$$H(Y|X) = -\sum_j p(x_j) \sum_i p(y_i|x_j) \log_2 p(y_i|x_j) \quad (8)$$

The amount by which the entropy of Y decreases reflects the additional information about Y provided by X and is called the information gain. Information gain is given by

$$IG(Y|X) = H(Y) - H(Y|X) = H(X) - H(X|Y) \quad (9)$$

Information gain is a symmetrical measure that is the amount of information gained about Y after observing X is equal to the amount of information gained about X after observing Y. By taking attributes as X and class as Y, we can rank attributes as per their influence or information gain over class.

IV. EXPERIMENT AND RESULTS

A. Dataset Description.

The experiment is conducted on KDDCup'99 dataset. The DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly as one specific attack type.

The dataset contain a total of 42 attributes including the attack type that is class label. All attributes are either categorical or continuous. Detail description of the dataset can be found at [17]. There are 22 types of attacks that are grouped into four main types tabulated in Table I.

Table 1. Category of Attacks in KDDCUP99 Dataset

Category	Class label(attack) in dataset
DOS-Denial of service	back,land, pod, neptune, smurf, teardrop
R2L-Remote to local	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient
U2R-User to root	Buffer_overflow, loadmodule, perl, rootkit
Probe	Ipsweep, nmap, portsweep, satan

Table II and Table III shows here the distribution of records in different classes for training and testing data set used in experiments.

Table 2. Distribution for Training Set

Class	No. Of Samples
probe	4107
dos	391458
U2r	52
R2l	1126
normal	97277
Total	494020

Table 3. Distribution for Testing Set

Class	No. of Samples
probe	1042
dos	65776
U2r	35
R2l	334
normal	23872
Total	91059

B. Evaluation Measurement

An Intrusion Detection System (IDS) requires high

accuracy and detection rate. Along with accuracy also intrusion detection system should provide very low false alarms rate ideally.

In general, the performance of Intrusion Detection System is measured and evaluated in term of accuracy, detection rate, and false alarm rate as in the following formula:

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (10)$$

$$\text{Detection Rate} = (TP) / (TP+FP) \quad (11)$$

$$\text{False Alarm} = (FP) / (FP+TN) \quad (12)$$

Table 4. Confusion Matrix for Evaluation

		Predicted Class	
		+	-
Actual Class	+(Normal)	True Negative (TP)	False Positive (FP)
	-(Intrusion/Attack)	False Negative (FN)	True Positive (TP)

True Positive (TP): attack occurs and detected

False Positive (FP): normal record predicted as attack

True Negative (TN): normal record predicted as normal

False Negative (FN): attack predicted normal

Table IV shows the categories of data behavior in intrusion detection for binary category classes (Normal and Attacks) in term of true negative, true positive, false positive and false negative.

C. Results and Discussion.

We perform all experiment on 2.53 GHz intel core i3 machine with 4GB of RAM. We use Weka framework available as open source at [18, 19] in JAVA and use Eclipse SDK 3.7.0 for development.

We changed virtual memory setting of Java Virtual Machine and heap memory is increased for loading of eclipse so that it can handle the very huge size of KDDCUP'99 dataset used in experiment.

We use the InfogainAttributeEval and chisquareAttributeEval method with Ranker as search algorithm for Information Gain and Chisquare attribute selection. Table V, here represents the rank of the attributes obtained.

Table VI shows the selected attribute using CfsSubsetEval attribute estimator with BestFirst search heuristic.

Table 5. Ranking of Attributes Using Chisquare and Information Gain

Rank	Information Gain (with Ranker)	Chi-square (with Ranker)
1	src_bytes	src_bytes
2	count	service
3	service	dst_bytes
4	dst_bytes	count
5	dst_host_srv_diff_host_rate	dst_host_diff_srv_rate
6	logged_in	dst_host_srv_diff_host_rate
7	srv_count	Srv_count
8	dst_host_count	dst_host_srv_count
9	dst_host_srv_count	Srv_diff_host_rate

For our experiment, we selected attribute set based on the repetition of attribute under the above three scheme. Our 8 attribute set for experiment with repetition is : service(3), count(3), src_bytes(3), dst_host_count(2), srv_count(2),dst_host_srv_count(2),dst_host_diff_srv_rate(2), logged_in(2).

We perform Naïve Bayes classification with all 41 attribute and also with our selected set of 8 attribute.

Table 6. Selected attributes for Correlation based Feature Selection

Correlation based feature selection (with BestFirst)
service
src_bytes
logged_in
iroot_shell
dst_host_diff_srv_rate
count
dst_host_count
dst_host_srv_diff_host_rate
srv_diff_host_rate

Next, we perform classification using Decision Table Majority (DTM) classifier using the CFS algorithm for attribute selection with BestFirst search approach. Initially starting with empty set in forward direction and then expanded up to 5 nodes in our experiment. Here the Subsets are evaluated using 10 fold cross validation on the training dataset. Following is the summary of results for DTM.

=== Summary ===

Number of training instances: 494020

Number of Rules: 1140

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 370

Merit of best subset found: 99.794

Evaluation (for feature selection): CV (leave one out)

Feature set: 5, 14, 22, 26, 35, 42

(src_bytes, Iroot_shell, guestlogin, srv_error_rate, dst_host_diff_srv_rate, label)

Decision table majority classifier is implemented and tested on our test dataset using output feature subset of CFS as a schema. Decision table majority classifier will generate the key using selected schema attributes and then insert records in the hash table with counter for each class using key generated. Decision table classifier searches for exact matches in the decision table using only the selected features in the schema.

Here if no instances are found then the class with majority in the decision table is returned as prediction. If matching is possible then the class with majority in the set of matching instances is returned.

Table VII shows the performance of the CFSDTM for individual category over various parameters like true positive rate, false positive rate, precision and recall. Categories here are four specific type of attacks that is probe, dos, u2r and r2l and also normal (records not classified as any kind of attack).

Table 7. Detailed Accuracy by Class

TP Rate	FP Rate	Precision	Recall	Class
0.992	0.001	0.913	0.992	probe
1	0.001	1	1	dos
0.714	0	1	0.714	u2r
0.844	0.001	0.86	0.844	r2l
0.995	0.001	0.997	0.995	normal

Table 8. Comparison of Detection Rate for Individual Class

		Classifying Scheme		
		NB with 41 attributes (DR %)	NB with 8 attributes (DR %)	CFSDTM with 5 attributes (DR %)
Category / Class	Probe	98.27	93.04	99.2
	dos	93.93	84.93	100
	u2r	62.85	51.65	71.40
	r2l	34.13	30.53	84.41
	normal	73.16	65.78	99.50

Table VIII here shows the comparison of detection rate for each individual class (either special types of attack or normal) under all the three classification schemes employed in the experiment.

Fig. 1 shows the graphical comparison of detection rate for attacks labeled probe and dos for the three schemes. Similarly Fig. 2 shows the comparison of detection rate for u2r and r2l kind of attack while Fig. 3 shows comparison for records that are not classified as any type of attack i.e. normal records under the three schemes in our experiment.

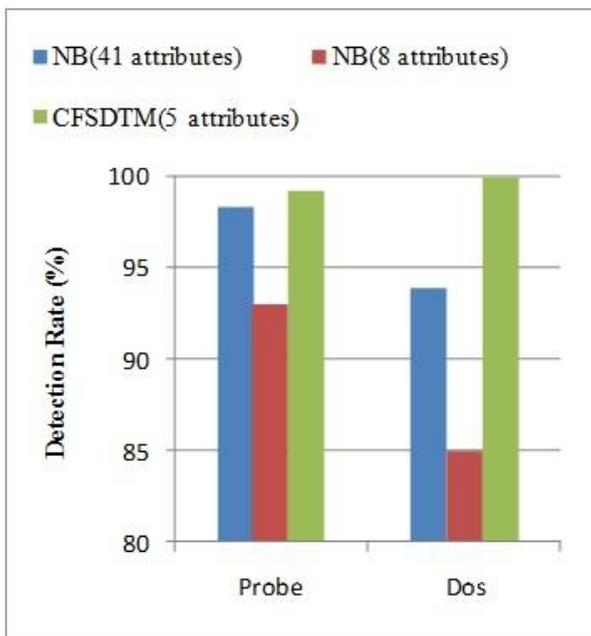


Fig. 1. Comparison of detection rate for probe and dos types of attack.

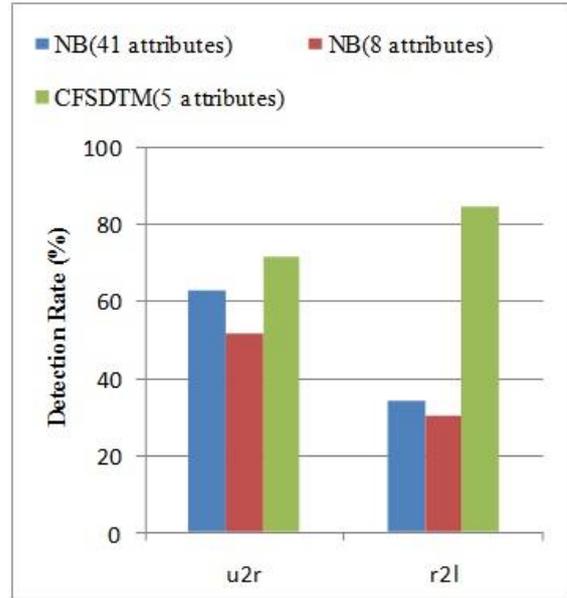


Fig. 2. Comparison of detection rate for u2r and r2l types of attack.

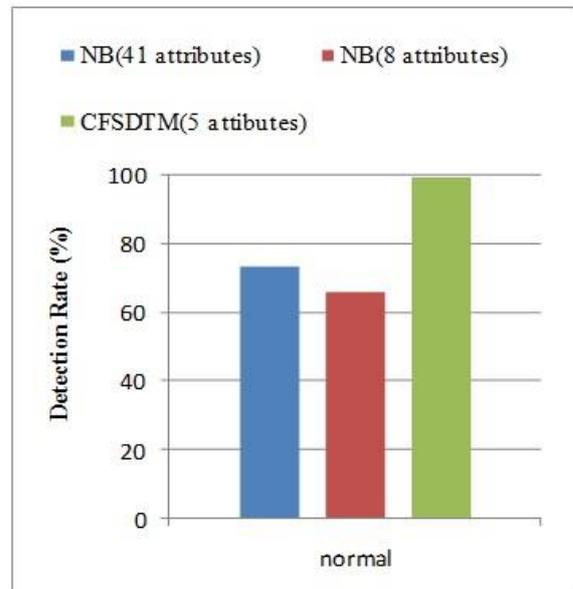


Fig. 3. Comparison of detection rate for normal records.

V. CONCLUSIONS & FUTURE WORK

In this paper, intrusion detection is performed on KDDCUP99 dataset. Intrusion detection is performed by naïve bayes on full set and reduced attribute set, derived from three attribute selection scheme based on majority in repetition. Results show considerable performance over reduced dataset also. We also perform DTM classification with CFS as attribute selection, because DTM is one of the powerful classifier due to exact match of attribute values that removes the strong independence assumption of naïve bayes. Results show very good performance with only 5 attributes also.

Future work includes testing the system with real world network data. The real time data can be generated and collected with the help of various attack simulation

tools that can cover variety of newly introduced attack definitions.

REFERENCES

- [1] Aleksanda Lazarevic, L. Ertoz, Aysel Ozgur, Jaideep Srivastava and Vipin Kumar, "A Comparative Study of Anomaly Detection Schemes in the Network Intrusion Detection", in Proceedings of Society for Industrial and Applied Mathematics, (SIAM) Conference on Data Mining, 2003.
- [2] Joseph Derrick, Richard W. Tibbs, Larry Lee Reynolds, "Investigating new approaches to data collection, management and analysis for network intrusion detection", Proceeding of the 45th annual south east regional conference, DOI = <http://dl.acm.org/citation.cfm?doid=1233341.1233392>, 2007.
- [3] Wenke Lee, Salvatore J. Stolfo and Kui W. Mok, "A Data Mining Framework for Building Intrusion Detection Model, Security and Privacy", Proceedings of the 1999 IEEE Symposium, pages 120-132, 1999.
- [4] E. Eskin, A. Arnold, M. Preau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data". Applications of Data Mining in Computer Society, Kluwer Academic Publishers, 2002.
- [5] Lee W., Stolfo S. and Mok K., "A Data Mining framework for Building Intrusion Detection Models", In Proceedings of the IEEE Symposium on Security and Privacy, 1999.
- [6] Luo J. and Bridges S. M., "Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection," International Journal of Intelligent Systems, (IJIS), John Wiley & Sons, Vol. 15, No. 8, pp. 687-704, 2000.
- [7] B. A. Nahla, B. Salem, and E. Zied, "Naive bayes vs decision trees in intrusion detection systems", In Proceeding of the ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004.
- [8] A. H. Sung, S. Mukkamala, "Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks", Symposium on Applications and the Internet, 2003.
- [9] Mukkamala S., Sung A.H. and Abraham A., "Intrusion Detection Using Ensemble of Soft Computing Paradigms", Third International Conference on Intelligent Systems Design and Applications, Springer Verlag Germany, pp. 239-248, 2003.
- [10] Hongjie Liu, Boqin feng, jianjie weng, "An Effective Data Classification Algorithm Based on the Decision Table", Seventh IEEE Association for Computer and Information Science(ACIS) International Conference on Computer and Information Science, 2008.
- [11] Jashan Koshal, Monark Bag, "Cascading of C4.5 Decision Tree and Support Vector Machine for Rule Based Intrusion Detection System", in International Journal of Computer Network and Information Security (IJCNIS), Vol. 4, pp 8-20, August 2012.
- [12] Ron Kohavi, "The power of decision Tables", in 8th European conference on Machine learning, pp.174-189, 1995.
- [13] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization", pp. 412-420, ICML, 1997.
- [14] H. Liu and, R. Setiono. Chi2, "Feature selection and discretization of numeric attributes, Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, pp. 338-391, 1995.
- [15] M. A. Hall, L. A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper", in Proceedings of Florida Artificial Intelligence Research Symposium, Orlando, FL, 1999, pp. 235-239.
- [16] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. 2nd edition, 2004.
- [17] KDD (1999). Available at <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [18] [http://weka.wikispaces.com/Eclipse/Eclipse+3.4.x+\(weka-src.jar\)](http://weka.wikispaces.com/Eclipse/Eclipse+3.4.x+(weka-src.jar)).
- [19] <http://www.cs.waikato.ac.nz/ml/weka/documentation.html>.

Authors' Profiles



Virendra Barot is currently working as an Assistant Professor at Information Technology Department, Sardar Vallabhbhai Patel Institute of Technology, Vasad., Gujarat, India. He has completed M. Tech in information Technology from IIT Roorkee, Roorkee, UttaraKhand in 2012 and B.E in information Technology from Dharmsinh Desai Institute of Technology, D.D.U State University, Nadiad, Gujarat, india. in 2003. He is having 11 Years of Experience in academics. His research areas include Cloud Computing, Intrusion Detection and Data Mining.



Sameer Singh Chauhan is currently working as an Assistant Professor at Information Technology Department, Sardar Vallabh bhai Patel Institute of Technology, Vasad. He has completed M.Tech. From IIT Roorkee, Roorkee (Uttarakhand), India in 2010 and B.E. from North Gujarat University, Patan, Gujarat, India. in 2001. He is having more than 12 Years of Experience in academics and research. His research areas are High Performance Computing, Cloud Computing, Grid Computing, Parallel Computing and Data Mining.



Bhavesh Patel is working as an Assitant Professor at Information Technology, Vasad. Gujarat, India. He has completed his M.E in Computer Science and Engineering from Government Engineering College, Gandhinagar in 2012 and B.E in Information Technology from S.V.I.T, Vasad, Gujarat, India. in 2004. He is having 10 years of experience in academics. His research areas include Information Security, Network Forensics and Data Mining.

How to cite this paper: Virendra Barot, Sameer Singh Chauhan, Bhavesh Patel, "Feature Selection for Modeling Intrusion Detection", IJCNIS, vol.6, no.7, pp.56-62, 2014. DOI: 10.5815/ijcnis.2014.07.08