2015 International Conference on Communication, Information & Computing Technology (ICCICT), Jan. 16-17, Mumbai, India

Improving Classification Using Preprocessing and Machine Learning Algorithms on NSL-KDD Dataset

Datta H.Deshmukh
Department Of Computer Engineering
Ramrao Adik Institute Of Technology
Navimumbai,INDIA
deshmukh.datta7@gmail.com

Tushar Ghorpade
Department Of Computer Engineering
Ramrao Adik Institute Of Technology
Navimumbai,INDIA
tushar.ghorpade@gmail.com

Puja Padiya
Department Of Computer Engineering
Ramrao Adik Institute Of Technology
Navimumbai,INDIA
puja .padiya@gmail.com

***Abstract*— Classification is the category that consists of identification of class labels of records that are typically described by set of features in dataset. The paper describes a system that uses a set of data pre-processing activities which includes Feature Selection and Discretization. Feature selection and dimension reduction are common data mining approaches in large datasets. Here the high data dimensionality of the dataset due to its large feature set poses a significant challenge. In Pre-processing with the help of Feature selection algorithm the various required features are selected, these activities helps to improve the accuracy of the classifier. After this step various classifiers are used such as Naive Bayes, Hidden Naive Bayes and NBTree. The advantage of Hidden Naive Bayes is a data mining model that relaxes the Naive Bayes Method's conditional Independence assumption. Also the next Classifier used is NBTree which induces a hybrid of decision tree classifiers and Naïve Bayes classifiers which significantly improves the accuracy of classifier and decreases the Error rate of the classifier. The output of the proposed method are checked for True positive, True negative, False positive, False negative. Based on these values the Accuracy and error rate of each classifier is computed.**

Keywords – classification, Feature selection, discretization, Naïve Bayes, Hidden Naïve Bayes, NBTree.

I. INTRODUCTION

The increasing number of threats against and vulnerabilities of a diverse set of targets, such as military, government and commercial network systems, require increasing situational awareness and various cyber security measures [1][14].

Intrusion detection system is a type of security management system for computers and networks. An Intrusion Detection system gathers and analyzes information from various areas Within a computer or a network to identify possible security breaches. The intrusion detection systems are a critical component in the network security. Data mining techniques [2][8] are used to explore and analyze large dataset and find useful patterns. Classification [16] is the category that consists of identification of class labels of records that are typically described by set of features in dataset.

The aim of the paper it to develop a system which uses various preprocessing methods such as Feature Selection [9] [13] and Discretization. With the help of Feature selection algorithm required features are selected and due to Discretization the data is discretized which can be applied to various classifier algorithms such as Naive Bayes[10], Hidden Naive Bayes and NBTree.

II. RELATED WORK

This paper present a literature review of few areas that covers span of research .The data set is publicly available for researchers through the website.

MahbodTavallaee, EbrahimBagheri, Wei Lu, and Ali A. Ghorbani [7] conducted a statistical analysis on this data set; they found some important issues which highly affects the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, they have proposed a new data set, NSL-KDD [3] which has the following advantages over the original KDD data set [11]:

1. It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
2. There is no duplicate records in the proposed test sets; therefore, the performance of the learners are not biased by the methods which have better detection rates on the frequent records.
3. The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
4. The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

Adetunmbi A. Olusola, Adeola S. Oladele and DaramolaO.Abosede [4] presented the relevance of each feature [6] in KDD 99 intrusion detection dataset to the detection of each class. Rough set degree of dependency and dependency ratio of each class were employed to determine the most discriminating features for each class. Selecting the right features is challenging, but it must be performed to reduce the number of features for the sake of efficient processing speed and to remove the irrelevant, redundant and noisy data for the sake of predictive accuracy.

III. DATASET

The NSL KDD Dataset [3] is one of the few currently available public datasets. The majority of the experiments in the intrusion detection domain are performed on this dataset. Since our model is based on supervised learning methods, NSL KDD is the available dataset that provides labels for both training and test sets. The study sample was created based on the 1998 DARPA intrusion detection evaluation offline dataset developed by the MIT Lincoln laboratory. The NSL KDD dataset is the public dataset on network events that contains a comprehensive set of labeled intrusion events. This dataset is quite large in terms of both number of instances and number of features, and it provides interesting characteristics on the distribution of events and on the dependencies between features. These interesting characteristics and challenges of the dataset make it much more appropriate for use as a benchmark in intrusion detection studies.

The dataset contains training data that include seven weeks of network traffic in the form of Transmission Control Protocol (TCP) dump data consisting of approximately 5 million connection records, each of which is approximately 100 bytes. The test data included two weeks of traffic, with approximately 2 million connection records. The 10% NSL KDD99 dataset was used as the training dataset in the competition. Each Connection record contains 7 discrete and 34 continuous features for a total of 41 features.

IV.PROBLEM DEFINITION

This Section is to define the problem definition associated with Intrusion Detection System in Data Mining. The Dataset for the experiments is NSL KDD Dataset which is one of commonly used public dataset in Intrusion Detection domain here the proposed system helps to Increase the accuracy and decreases the error rate of classifiers but The problem occurred here is the High dimensionality of the dataset which effects the accuracy of classifiers. So the solution for this problem is given in the proposed system by applying the ideal feature selection algorithm i.e. Fast Correlation Based Filter (FCBF) algorithm which reduces the dimensionality of the dataset in pre-processing part as well as Here The selection of proper algorithm for classifier also plays very important role. For this in the Proposed system the Hidden Naïve Bayes & NBTree algorithms are used which relaxes the Naive Bayes methods conditional independent assumption and helps to increase the accuracy of the classifier.

V. THE PROPOSED METHOD

The proposed method uses the NSL KDD'99 dataset [3] for the intrusion detection for building the classification models by supervised training method. It consists of the following steps as Shown in Fig 1.

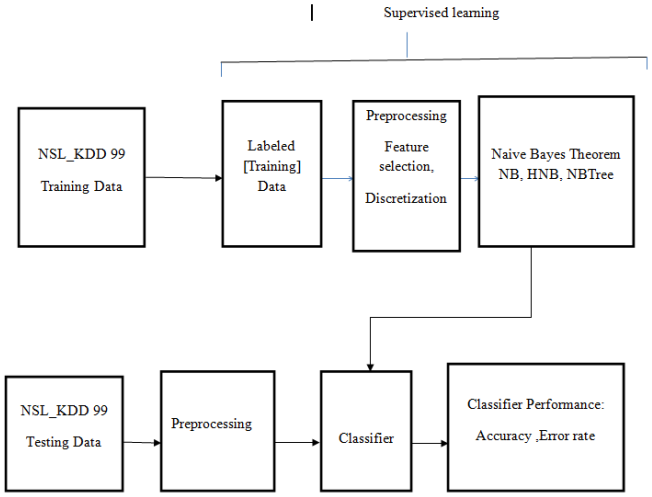

Fig.1 Proposed System

A) Component Details of the Proposed System

Step 1) Split Dataset into two separate sets i.e. training set and testing set. Here the Dataset consists of Training set and Testing set.

Step 2) Perform Pre-processing. This step includes a Feature selection model based on Fast Correlation Based Filter method. Feature selection, as a pre-processing step to machine learning, is effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. This method can identify relevant features for further processing.

Step 3) Perform Final pre-processing task which includes Discretization. Although the HNB classifier model and NBTree classifier model are based on discrete features, the NSL KDD99 dataset mainly consists of continuous features, which need to be first converted to discrete features. Here Equal Width Discretization Technique is implemented.

Step 4) Various Classifiers [15] are implemented such as Naive Bayes, Hidden Naive Bayes, NBTree and Different parameters are calculated such as TP rate, FP rate, TN rate, FN rate, and Accuracy and Error rate.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

B) Data Pre-Processing

a) Feature Selection

Feature selection [13] is a process of selecting a subset of relevant features by applying certain evaluation criteria. In general, feature selection process consists of three phases. It starts with selecting a subset of original features and evaluating each features worth in the subset. Secondly, using this evaluation, some features in the subset may be eliminated or enumerated to the existing subset. Thirdly, it checks whether the final subset is good enough using certain evaluation criterion. This approach removes redundant or irrelevant features from the dataset to prevent decreases in classification accuracy and unnecessary increases in computational costs.

This project has implemented FCBF algorithm [18]. This Fast Correlation Based Filter (FCBF) is a filter model feature selection algorithm that measure Feature- class and feature feature correlation .FCBF starts by selecting a set of features s that are highly correlated to the class and less correlated to other feature.

b)Discretization

Discretization is the process of converting the continuous domain of a feature into a nominal domain with a finite number of values. Front-end discretization might be necessary for some classifiers if their algorithms cannot handle continuous features by design. Additionally, earlier studies showed that discretization improves the accuracy of classifiers, including naïve Bayes classifiers, especially in larger datasets.Numerous studies have examined discretization methods in the last two decades to determine how continuous values should be grouped, how cut points should be positioned on the continuous scale, and how many intervals should be used to generate datasets.

In this study, EWD i.e Equal Width Discretization technique will be used. This method is selected because of their performance on large datasets, particularly the KDD'99 dataset. It is the simplest method to discretize a continuous-valued attribute by creating a specified number of bins. The bins can be created by equal-width.

It divides the range into N intervals of equal size: uniform grid .if A and B are the lowest and highest values of the attribute, the width of intervals will be as shown in Equ.1.

$$W = \frac{(B-A)}{N} \qquad (1)$$

VI. ALGORITHMS:

A) Naïve Bayes

A Naïve Bayes (NB) classifier is a simple probabilistic classifier based on Bayes theorem where every feature is assumed to be class-conditionally independent. In naïve Bayes learning, each instance is described by a set of features and takes a class value from a predefined set of values. Classification of instances gets difficult when the dataset contains a large number of features and classes because it takes enormous numbers of observations to estimate the probabilities. When a feature is assumed to be class-conditionally independent, it really means that the effect of a variable value on a given class is independent of the values of other variables.

B) Hidden Naïve Bayes

An extended version of the Naive Bayesian classifier is the hidden naive Bayes (HNB) classifier [19][20], which relaxes the conditional independence assumption imposed in the Naive Bayesian model. The HNB model relies on the creation of another layer that represents a hidden parent of each attribute. The hidden parent combines the influences from all of the other attributes .The structure is as shown in Fig.2.

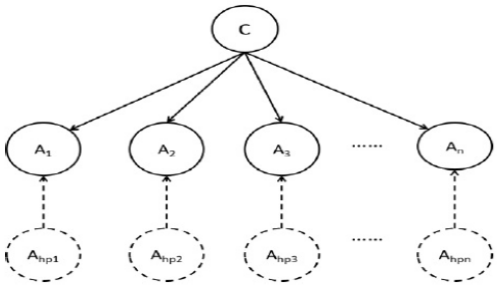

Fig 2.HNB Structure

In the HNB model, each attribute Ai has a hidden parent Ahpi,where i = 1, 2,, n represents the weighted influences from all of the other attributes, as shown with the dashed circles.

C) NBTREE

NBTree [17] which induces a hybrid of decision-tree classifiers and Naive Bayes classifiers: the decision-tree nodes contain univariate splits as regular decision-trees[5],but the leaves contain Naive-Bayesian classifiers. The algorithm is similar to the classical recursive partitioning schemes except that the leaf nodes created are Naive Bayes categorizers instead of nodes predicting a single class.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

VII.REULTS

This section gives the details of the technical platform required for conducting the experiments. In this work we will be using JAVA Net Beans IDE for Implementation of various algorithms defined in the project .MySQL will be used as a back end so as to facilitate the use of substantial algorithm and additional API is used WEKA.jar.we first observed the results obtained by the experiments performed on the NSL KDD Dataset for Intrusion detection problem by using feature selection method. We obtained the output which is reduced set of features as shown in fig 3.

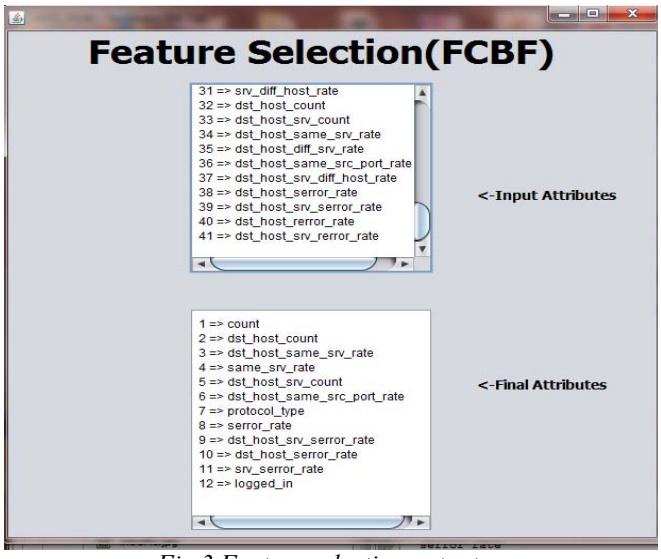

Fig 3.Feature selection output

We applied this output set to the first classifier i.e. Naive Bayes and computed various classifier performance evaluation measures such as True Positive Rate, False Positive Rate, True Negative Rate, False Negative Rate and computed Accuracy and Error Rate as important performance measures. Accuracy is the fraction of correctly classified instances, and error rate is the fractions of misclassified instances in a dataset. These two measures effectively summarize the overall performance of the classifier.

Various combinations are done to improve the accuracy of classifier [12]such as After CFS,After CFS + Discretization,CFS + Discretization + Hidden Na¨ıve Bayes,CFS + Discretization +NBTree.

a) After CFS: The Input to the feature selection algorithm is the entire dataset (41 features) and Output is (12 features). So In this case Naive Bayes classifier after feature selection shows the Accuracy as 87.04% and Error Rate as 12.96%. The Accuracy is obtained by observing the True Positive Rate, False Positive Rate, True Negative Rate, False Negative Rate for Normal and Anamoly class as shown in following Table 1.

Table 1 .Classifier performance parameters

Parameters	Number of Instances
False Negative of Normal (FN Rate)	138
False Positive of Normal (FP Rate)	510
True Negative of Normal (TN Rate)	1876
True Positive of Normal (TP Rate)	2476
False Negative of Anamoly (FN Rate)	510
False Positive of Anamoly (FP Rate)	138
True Negative of Anamoly (TN Rate)	2476
True Positive of Anamoly (FN Rate)	1876

Now put all values in the following formula to check Accuracy and Error Rate

1. Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

   = (2476 + 1876) / (2476 + 1876 + 510 + 138)

   = 4352 / 5000

   Accuracy = 0.8704 (87.04%)

2. Error Rate = 1 - Accuracy

   = 1 - 0.8704 = 0.1296 (12.96%)

b) After CFS + Discretization

The features selected out of 41 features available in the dataset based the combination of these feature selection and discretization method. So here various combinations are checked such as case 1) CFS + Discretization +Naive Bayes, case 2) CFS + Discretization + Hidden Naive Bayes, case 3) CFS + Discretization +NBTree.

Table 2: Accuracy and Error Rate Of Classifier

Classifier	Accuracy	Error Rate
Naïve Bayes	88.20%	11.80%
Hidden Naïve Bayes	93.40%	6.60%
NBTree	94.60%	5.40%

After performing first combination i.e case 1) CFS + Discretization + Naive Bayes we computed various classifier performance evaluation measures such as True Positive Rate, False Positive Rate, True Negative Rate, False Negative Rate and computed Accuracy and Error Rate as important performance measures.Here we got the Accuracy as 88.20% and Error Rate as 11.80%.

In case 2) i.e CFS + Discretization + Hidden Na¨ıve Bayes .the classifier performance has increased in terms of Accuracy. Accuracy is increased from 88.20% to 93.40% and Error Rate is decreased from 11.80 to 6.60 as shown in Table 2 .

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

In case 3) i.e CFS + Discretization + NBTree ,the classifier performance has increased in terms of Accuracy.Accuracy is increased from 93.40% to 94.60% and Error Rate is decreased from 6.60 to 5.40% as described in Table 2.

The graph of Accuracy and Error Rate of various classifiers such as Naive Bayes, Hidden Na¨ıve Bayes and NBTree for various combinations such as case 1) CFS + Discretization + Na¨ıve Bayes, case 2) CFS + Discretization + Hidden Na¨ıve Bayes, case 3) CFS + Discretization + NBTree are shown in fig 4 and Fig.5.

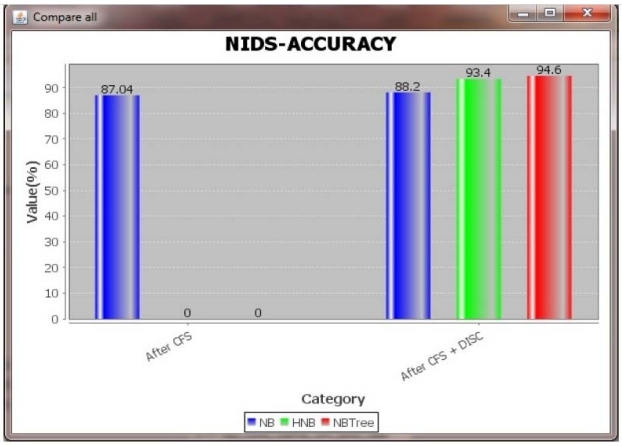

Fig 4. Accuracy Graph

Based on above test cases it is observed that NBTree algorithm perform well in terms of Accuracy and Error Rate as Compare to Traditional Na¨ıve Bayes. The graphs for the above discussion are shown in fig 4 and fig.5.

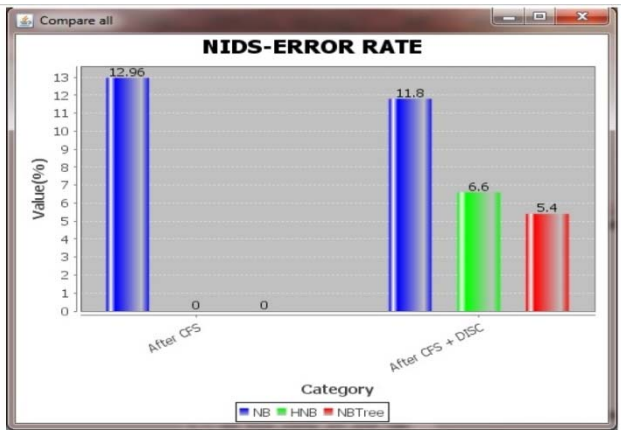

Fig 5. Error Rate Graph

Classifier performance evolution parameters

1. TP RATE: True Positive Rate is the number of actual positives classified correctly as true. It is ratio of positive instances that were correctly identified to the total number of actual positive instances.
2. FP RATE: is the number of actual negatives classified as positives. It is ratio of negative instances that were incorrectly classified as positive to the total number of negative Instances.
3. TN RATE: True Negative Rate is the number of actual negatives correctly classified as negatives.
4. FN RATE: False Negative Rate is the number of actual positives incorrectly classified as negatives.
5. Accuracy: Accuracy is the fraction of correctly classified instances. Accuracy is the number of correct classifications divided by the total number of classifications.
6. Error Rate: Error rate is 1 minus accuracy. The number of misclassified instances.
7. Confusion Matrix : The Confusion Matrix Report is useful for validating classification models. The diagonal elements are the counts of the correct predictions and the off diagonal elements the incorrect predictions is for a binary classifier that is the simplest classifier form. Note that binary classes can be based on yes / no or attack / no attack in an intrusion detection context.

VIII.CONCLUSION

The proposed system explains the need to apply data mining methods to network events to classify network attacks and improve the accuracy of classifier. The System has more focus on how to increase the accuracy of the classifier. for this purpose the project has implemented various preprocessing steps on the existing NSL-KDD Dataset such as Feature selection and Discretization .The proposed system attempts to overcome the problem of High Dimensionality of the Dataset by selecting the proper feature selection algorithm such as Fast Correlation based Filter Algorithm .the next important issue is to section of proper algorithm for classifier so in the existing system Na¨ıve Bayes has drawback of conditional independence assumption. To overcome this issue the project has implemented various classifiers such as Hidden Naïve Bayes Classifier and NBTree classifier. After implementation the proposed system has improved the accuracy of the classifier and decreased the Error rate.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE

REFERENCE

[1] Kabiri, Peyman, and Ali A. Ghorbani. "Research on Intrusion Detection and Response: A Survey." IJ Network Security 1.2 (2005): 84-102.

[2] Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., & Tan, P. N. (2002, November).Data mining for network intrusion detection. In Proc. NSF Workshop on Next Generation Data Mining (pp. 21-30)

[3] The NSL KDD Dataset. [Online]. Available http://nsl.cs.unb.ca/NSL-KDD/,On Dated July 30, 2013.

[4] Olusola, Adetunmbi A., Adeola S. Oladele, and Daramola O. Abosede. "Analysis of NSL KDD'99 Intrusion Detection Dataset for Selection of Relevance Features." Proceedings of the World Congress on Engineering and Computer Science.Vol. 1. 2010.

[5] Kumar, Manish, M. Hanumanthappa, and TV Suresh Kumar. "Intrusion Detection System using decision tree algorithm." Communication Technology (ICCT), 2012 IEEE 14th International Conference on. IEEE, 2012.

[6] Kayacik, H. Gnes, A. NurZincir-Heywood, and Malcolm I. Heywood. "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets." Proceedings of the third annual conference on privacy, security and trust. 2005.

[7] Tavallaee, Mahbod, et al. "A detailed analysis of the KDD CUP 99 data set." Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009.

[8] Lee,Wenke, Salvatore J. Stolfo, and KuiW. Mok. "A data mining framework for building intrusion detection models." Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on. IEEE, 1999.

[9] Amudha, P., and H. Abdul Rauf. "Performance Analysis of Data Mining Approaches in Intrusion Detection." Process Automation, Control and Computing (PACC), 2011 International Conference on. IEEE, 2011

[10] Panda, Mrutyunjaya, and ManasRanjanPatra. "A comparative study of data mining algorithms for network intrusion detection." Emerging Trends in Engineering and Technology,2008. ICETET'08. First International Conference on. IEEE, 2008.

[11] KDD-Cup. (1999)., from http:// kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,KDD Cup 1999 Data Retrieved July 29, 2013.

[12] Bolon-Canedo, Vernica, N. Sanchez-Maroo, and Amparo Alonso-Betanzos. "A combination of discretization and filter methods for improving classification performance in KDD Cup 99 dataset," Neural Networks, 2009. IJCNN 2009. International Joint Conferenceon. IEEE, 2009.

[13] Bolon-Canedo, Veronica, Noelia Sanchez-Marono, and Amparo Alonso-Betanzos. "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset." Expert Systems with Applications 38.5 (2011): 5947-5957.

[14] Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert systems with Applications, 29(4), 713-722.

[15] Nguyen, HuyAnh, and Deokjai Choi. "Application of data mining to network intrusion detection: classifier selection model," Challenges for Next Generation Network Operations and Service Management. Springer Berlin Heidelberg, 2008. 399-408.

[16] Beniwal, Sunita, and Jitender Arora. "Classification and feature selection techniques in data mining." International Journal of Engineering Research and Technology. Vol. 1. No. 6 (August-2012). ESRSA Publications, 2012.

[17] Kohavi, Ron. "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." In KDD, pp. 202-207. 1996.

[18] Yu, Lei, and Huan Liu. "Feature selection for high-dimensional data: A fast correlation based filter solution." In ICML, vol. 3, pp. 856-863. 2003.

[19] Zhang, Harry, Liangxiao Jiang, and Jiang Su. "Hidden naive bayes." Proceedings of the National Conference on Artificial Intelligence. Vol. 20. No. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[20] Levent Koc , Thomas A. Mazzuchi, ShahramSarkani, A network intrusion detection system based on a Hidden Na¨ıve Bayes Multiclass classifier, Expert Systems with Applications,
2012.

978-1-4799-5522-0/15/$31.00 ©2015 IEEE