Supervised Versus Unsupervised Discretization for Improving Network Intrusion Detection

Doaa Hassan
Computers and Systems Department
National Telecommunication Institute
Cairo, Egypt
Email: doaa@nti.sci.eg

***Abstract*—Discretization acts as an important preprocessing step in the data mining process that transforms the continuous values of data features into discrete ones. Machine learning (ML) algorithms such as Support Vector Machines (SVM) and Random Forests (RF) have been widely known for their robustness to the dimensionality of the data and hence they are used for classification of high-dimensional data. Moreover, other ML algorithms such as Naive Bayes (NB) basically use feature discretization and also developed for classifying high dimensional data. This paper investigates the effect of pre-processing the dataset with either supervised or unsupervised discretization techniques on improving the performance of the three aforementioned ML algorithms. Those algorithms run on NSL-KDD, a high dimensional dataset and an improved version of KDD cup 99 dataset that is widely used for network intrusion detection. Our results show that preprocessing NSL-KDD with either supervised or unsupervised discretization techniques does not improve the performance of RF. On the other hand the supervised entropy-based discretization (EBD) and unsupervised Proportional k-Interval Discretization (PKID) lead to a little better performance of SVM and a significant improvement in the performance of NB. Also the results show that the performance of NB with discretization is clearly less than the performance of SVM and RF either with or without discretization. Therefore the paper proposes an approach that combines discretization with the wrapper feature selection method in order to improve the performance of NB with discretization and make it so closed to the performance of either SVM or RF.**

***Index Terms*—network intrusion detection; data discretization; wrapper feature selection; classification**

I. Introduction

Since the network attacks have increased in number and severity over the past few years, network intrusions detection is increasingly becoming a critical component to secure the computer network. Recentely, optimizing the performance of intrusion detection using data mining techniques has received more attention from the research community [1] due to the huge volumes of network audit data analyzed to detect the complex and dynamic properties of intrusion behaviors. One of those techniques is to preprocess the network audit data using discretization algorithms [2] in order to transform the continues features of such dataset into discrete ones by creating a set of disjoint intervals. Such an approach is mainly used to increase the capability of classifier to successfully distinguish between the anomalous network traffic and the normal one. A two common existing mechanisms for discretization are: the supervised method which uses the class information and the unsupervised one that does not use it at all [7], [3].

In this paper, we investigate the application of the two aforementioned discretization mechanisms to a high dimensional dataset that have been widely used for intrusion detection: The NSL-KDD dataset [5] which is an improved version of KDD cup 99 dataset [4]. The later includes a wide variety of intrusions simulated in a U.S military network environment. The former consists of reasonable number of records selected from the complete KDD dataset in order to enhance the performance of prediction. We study the effect of both discretization methods on improving the performance of three ML supervised algorithms including the Support Vector Machines (SVM) [26], Random Forests (RF) [27] and Naive Bayes (NB) [25] that run on NSL-KDD dataset. Those classifiers are used for detecting anomalous network connections in this dataset. Moreover, they are widely known for their robustness to the curse of dimensionality problems and hence they are used for the classification of high-dimensional data [6]. Our experimental results show that applying either supervised or unsupervised discretization to NSL-KDD dataset as pre-processing step before classification does not to lead to improve the performance of RF on average, while it achieves in average a little better improvement in the performance of SVM and a notable improvement in the performance of NB, particularly using either the supervised EBD or the unsupervised PKID methods. Finally, the results show that though applying discretization to a dataset before classification leads to a significant improvement in the performance of NB that runs on the discretized dataset, the performance of NB with discretization is clearly still less than the performance of either SVM or RF either with or without discretization. This motivates us to propose a new approach in this paper that improves the performance of NB classifier with discretization. This approach relies on combining discretization with the wrapper feature selection method, and then applying such combination to a dataset that has been preprocessed by discretization in order to get a reduced version of it. Next the NB is applied to the obtained reduced discretizated version of the dataset. Our experimental results show that the performance of NB using the proposed approach outperforms the performance of SVM without discretization and is very closed to the performance of either SVM or RF with discretization.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

The structure of this paper is organized as follows: In Section II, we provide a background on supervised and unsupervised discretization techniques. In Section III, we introduce our experimental settings. In Section IV, we provide an empirical evaluation, by evaluating the performance of three supervised classifiers for identifying network intrusions: the SVM, RF and NB. Those classifiers run on either the original NSL-KDD data set or various versions of it that are preprocessed with either supervised or unsupervised discretization techniques. In Section V, we propose a methodology that relies on combining discretization with wrapper method for feature selection in order to improve the performance of NB classifier with discretization. In Section VI, we present the related work. Finally we conclude the paper in Section VII with some directions for future work.

II. Background

A. Supervised Discretization

Supervised discretization methods [3] rely on using any information in the target variable (i.e., class attribute) for performing a discretization of continuous attributes into disjoint (discrete) intervals. This is done by finding the proper/meaningful intervals caused by cut-points. Different research methods have been proposed in the literature for performing supervised discretization of continuous features including the error-based method [8], entropy-based method [10], and the statistics-based method [9]. The basic idea in all of these methods is to determine whether intervals are selected using metrics based on error in the training data, entropy of the intervals, or some statistical measures.

In this paper we use the entropy-based discretization (EBD) proposed by Fayyad and Irani [10] for performing the supervised discreitization as it is the most common supervised discretization technique and it has been shown that it outperforms the statistics and error based methods [8]. This method is based on using the entropy measure as a standard criterion for recursively partitioning the values of a continuous feature and Minimum Description Length (MDL) principle [11] as a criterion for stopping this partitioning.

B. Usupervised Discretization

The unsupervised discretizations methods are preferred when no class information is avilable. They rely on dividing the continuous ranges of each discretizated attribute into subranges according to a parameter specified by the user (e.g., range of values or number of instances in each interval).

There are two main representative algorithms of unsupervised discretizations: the equal width binning (EWB) and the equal-frequency binning (EFB). The EWB algorithm is the simplest one and it determines the minimum and maximum values of the discretized attribute in order to determine its range. Next it divides that range into equal width discrete intervals (k equal sized bins), where K is a parameter determined by the user. The EFB algorithm determines the minimum and maximum values of the discretizated feature, then sorts all of those values in ascending order. Next, it divides the range of those values into a set of intervals (i.e., k bins, where k is also a parameter determined by user), where every interval/bin contains the same number of sorted values.

Proportional k-Interval Discretization (PKID) is another unsupervised discretization method that was introduced in [12], [13] as an improvement to EDB and EFB discretization methods in order to improve the performance of NB classifier. PKID tunes the number and size of interval in order to adjust the NB probability estimation bias and variance as a mean to decrease the classification error. In summary, the basic idea of PKID is that it relates the discretization bias to the interval size and discretization variance to the interval number. Therefore, larger the interval size indicates the lower the variance but the higher the bias. In contrast, the smaller the interval size indicates the lower the bias but the higher the variance. Thus by tuning the interval size and number, a good trade-off between the bias and variance can be found and hence lower learning error can be achieved.

III. Experimental Settings

A. Experimental Environment

We have run our experiment on a windows laptop machine with 2.6 GHZ processor Intel core (TM)i5 and 4 G Memory Rams. We have used Weka [19], a free open source software data mining tool for performing discretization, feature selection and classification. Weka also provides different types of filters [20] and it is a rich suite of several machine learning algorithms. Using Weka filters allows to create different subsets of the dataset in which the overall class distribution in the original dataset is retained. Moreover, it allows easily to perform either supervised or unsupervised discretizations of continuous features in the dataset before either doing feature selection or applying classification algorithms.

B. NSL-KDD Dataset

NSL-KDD [5] is a reduced version of the original KDD cup 99 dataset [4]. It consists of 125973 network connections records. 67343 of them are normal, while the remaining 58630 records are anomaly. NSL-KDD records were selected from the complete KDD Cup 99 dataset in order to enhance the performance of prediction by considering the following differences over the original KDD 99 dataset:

- It does not include redundant records in the training set, so the classifiers will not be biased towards more frequent records.
- There are no duplicate records in the proposed test sets.

NSL-KDD has the same features as KDD Cup 99 in addition to a class attribute that has 2 possible values: normal and anomaly. Either KDD cup or NSL-KDD consists of 41 features including 34 numerical attributes and 7 nominal ones. Those features address some parameters for each network connection. Each anomaly connection has an attack pattern that falls into one of four possible categories: Denial of Service Attack (DOS), Users to Root Attack (U2R), Remote to Local Attack (R2L) and Probing Attack (PROBE) [5].

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Since NSL-KDD is massive, we have used the re-sampling mechanism [21] for creating three subsamples of 10%, 20% and 30% of the original dataset for carrying out the empirical evaluation. In each of those subsample, the overall class distribution in the original dataset is retained.

C. Discretization methods and their application

We have applied four discretization methods. The first is the Entropy Based Discretization (EBD) which is a supervised discretization method. The other three methods are unspervised discretization methods. Two of them are classic unsupervised discretization algorithms which are the Equal Width Bining discretization (EWBD) and Equal Frequency Bining Discretization (EFBD). The third one is the Proportional k-Interval Discretization (PKID) that is considered as an improvement to EWBD and EFBD methods and which works well with large datasets. Each of the four discretization methods is applied to three sub-sets of NSL-KDD dataset before applying classification to those dataset subsets to predict the anomaly network connections. Therefore, for each dataset sub-set analyzed, we have five versions: the original dataset sub-set and other four versions of it resulting from applying each of the four aforementioned discretization methods to this dataset sub-set. Since we have used three sub-sets of NSL-KDD dataset, this lead to the creation of 15 datasets.

D. Supervised learning

We have applied three machine learning algorithms that are more convenient for high dimensional data in addition to their ability to handle both discrete continuous features, namely, RF, SVM and NB. More precisely, RandomForest, SMO and NaiveBayes implementations in Weka respectively. Each version of the dataset sub-subset has been used in 10-fold cross validation, a standard approach for evaluating the performance of classification algorithms (where 90% of the dataset samples is randomly selected for training and the remaining 10% is kept for testing).

IV. Supervised versus Unsupervised Discretization: Empirical Evaluation

A. Performance Measures

We have evaluated the classification performance with accuracy, the false positive rate (FPR) and false negative rate (FNR). FPR measures the proportion of normal network connections that are erroneously classified as anomaly ones. On the other hand, FNR measures the proportion of anomaly network connections that are miss-detected. We choose to use FPR and FNR as measures for the performance of classifiers since they are common parameters that are widely used by researchers to describe the performance of an intrusion detection system (IDS) [22]. FPR, represents a Type I error that occurs when the IDS misidentifies a normal event as an attack. In contrast, FNR represents a Type II error that occurs when the IDS misidentifies the anomaly pattern or event as a normal one.

The accuracy, FPR and FNR are calculated by the following criteria:

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

$$FPR = \frac{TN}{FP + TN}$$

$$FNR = 1 - TPR$$

where TPR is the true positive rate that measures the proportion of normal network connections that are correctly identified. TPR is calculated by the following criterion:

$$TPR = \frac{TP}{TP + FN}$$

where,

- TP is the number of correctly classified network connections as normal ones.
- FP is the number of incorrectly classified normal network connections as anomaly ones.
- FN is the number of incorrectly classified anomaly network connections as normal ones.
- TN is the number of correctly classified network connections as anomaly ones.

B. Reuslts

Table I shows the accuracy, FPR and FNR calculated for each classifier when it runs on each original NSL-KDD dataset subset without applying discretization to it. It is clear from the table that RF achieves the best performance on the average followed by SVM then NB achieves the lowest performance.

TABLE I
THE ACCURACY, FPR AND FNR CALCULATED FOR EACH CLASSIFIERS WHEN IT RUNS ON EACH ORIGINAL NSL-KDD DATASET SUBSET WITHOUT DISCRETIZATION.

Dataset	Classifier	Accuracy	FPR	FNR
NSL-KDD-10%	SVM	97.42%	0.04	0.013
	RF	99.65%	0.006	0.001
	NB	90.93%	0.083	0.097
NSL-KDD-20%	SVM	97.32%	0.041	0.014
	RF	99.81%	0.003	0.001
	NB	89.59%	0.123	0.088
NSL-KDD-30%	SVM	97.33%	0.04	0.014
	RF	99.84%	0.002	0.001
	NB	90.29%	0.13	0.068
Average	SVM	97.36%	0.04	0.014
	RF	99.77%	0.004	0.001
	NB	90.27%	0.112	0.084

Table II shows the accuracy, FPR and FNR calculated for each experimental classifier when it runs on versions of each dataset subset discretizated using either the supervised or unsupervised discretizataion techniques presented in this paper.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE II
THE PERFORMANCE OF EXPERIMENTAL CLASSIFIERS THAT RUN ON DIFFERENT DISCRETIZATED VERSIONS OF EACH NSL-KDD DATASET SUBSET.

Dataset	Discretizor	Classifier	Accuracy	FPR	FNR
NSL-KDD-10%	EBD	SVM	99.41%	0.009	0.003
		RF	99.68%	0.005	0.001
		NB	96.42%	0.068	0.007
	EWBD	SVM	97.91%	0.028	0.014
		RF	98.88%	0.016	0.007
		NB	92.63%	0.137	0.018
	EFBD	SVM	99.46%	0.007	0.004
		RF	99.65%	0.006	0.001
		NB	92.87%	0.139	0.011
	PKID	SVM	99.55%	0.005	0.004
		RF	99.56%	0.008	0.001
		NB	96.65%	0.061	0.009
NSL-KDD-20%	EBD	SVM	99.58%	0.006	0.002
		RF	99.76%	0.004	0.001
		NB	96.55%	0.064	0.009
	EWBD	SVM	98.04%	0.026	0.014
		RF	98.94%	0.015	0.007
		NB	92.74 %	0.137	0.017
	EFBD	SVM	99.54%	0.006	0.003
		RF	99.73%	0.005	0.001
		NB	92.98%	0.139	0.01
	PKID	SVM	99.66%	0.004	0.003
		RF	99.71%	0.005	0.001
		NB	96.60%	0.062	0.01
NSL-KDD-30%	EBD	SVM	99.73%	0.004	0.001
		RF	99.87%	0.002	0
		NB	96.38%	0.067	0.009
	EWBD	SVM	97.94%	0.027	0.014
		RF	99.09%	0.011	0.007
		NB	92.69%	0.136	0.018
	EFBD	SVM	99.64%	0.005	0.003
		RF	99.83%	0.003	0.001
		NB	92.58%	0.144	0.012
	PKID	SVM	99.73%	0.003	0.002
		RF	99.80%	0.003	0.001
		NB	96.65%	0.061	0.009

Table III shows the average accuracy, FPR and FNR calculated for each experimental classifier when it runs on discretizated versions of the three NSL-KDD subsets. Figure 1 shows the average accuracy of each classifier as presented in Table III, when it runs on different discretizated versions of each datset subset.

TABLE III
THE AVERAGE PERFORMANCE OF EXPERIMENTAL CLASSIFIERS THAT RUN ON DIFFERENT DISCRITIZED VERSIONS OF NSL-KDD DATASET SUBSETS.

	Discretizor	Classifier	Accuracy	FPR	FNR
Average	EBD	SVM	99.57%	0.006	0.002
		RF	99.77%	0.0037	0.0007
		NB	96.45%	0.0663	0.0083
	EWBD	SVM	97.96%	0.0270	0.0140
		RF	98.97%	0.014	0.007
		NB	92.69%	0.137	0.018
	EFBD	SVM	99.55%	0.0060	0.003
		RF	99.74%	0.0047	0.001
		NB	92.81%	0.14	0.011
	PKID	SVM	99.65%	0.004	0.003
		RF	99.69%	0.0053	0.001
		NB	96.63%	0.061%	0.009

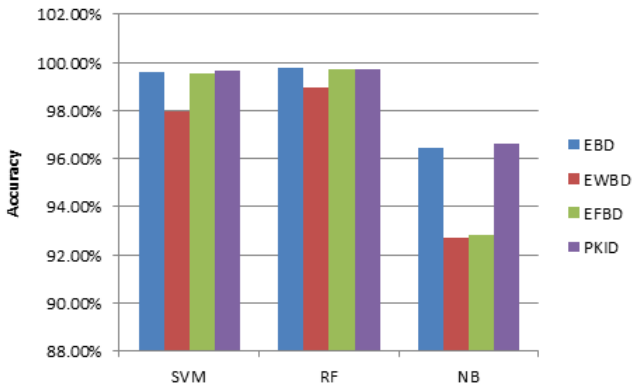

Fig. 1. The average accuracy of each classifier with different discrtized versions of each NSL-KDD subset.

Clearly from the figure we notice that each classifier achieves the best accuracy with either EBD or PKID then followed by EFBD and then EWBC. Moreover, by comparing the average performance of classifiers when run on the original NSL-KDD dataset (as shown in the Table I) to their average performance when they run on the discrtized versions of the original dataset (as shown in Tables III), we can conclude the following:

- Discretization does not lead to better performance with RF. However it leads to a little better performance with SVM and a significant improvement in performance with NB.
- Classification that runs on a dataset discretized with supervised EBD always outperforms the one that runs on

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

a dataset discretized with either unsupervised EWBD or EFBD. However it does not outperform the classification that runs on a dataset discretized with unsupervised PKID for either SVM or NB and it only outperforms it for RF.

- Though discretization leads to a significant increase in performance with NB, we notice that the performance of NB with either supervised or unsupervised discretization methods is still less than the performance of SVM and RF classifiers either with or without discretization. This motivates us to propose an approach to improve the performance of NB with discretization (as we will see in the next section).

V. How to improve the performance of Naive Bayes with Discretization

Our proposed approach to improve the performance of NB with discretization relies on applying feature selection by a combination of discretization and wrapper method to a dataset that has been preprocessed by discretization, then applying NB to the obtained reduced discretizated version of the dataset. The discretization of dataset is conducted before conducting feature selection with wrapper method to improve the performance of wrapper method in selecting the best subset of features that lead to the best performance [28]. Next the obtained subset of selected features is applied to a dataset that has been preprocessed by discretization. Then the obtained reduced discretized version of the dataset is used by NB classifier to perform classification. Figure 2 illustrates our proposed approach.

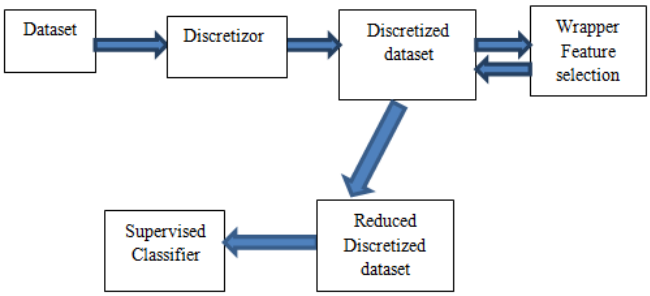

Fig. 2. Overview of the proposed approach for improving the performance of NB with discretizataion.

A. Feature selection with wrapper Method

Feature selection with wrapper method is a well known method for determining the most effective features of a dataset in data mining field [24]. It uses the predictor itself to evaluate the usefulness of features. The first step in feature selection by wrapper method is to use different search techniques such as best first Search, random search and depth first search or others to create all possible subsets of features from the feature vector. Then a predictor is induced from the features in each subset and the subset of features that leads to the optimum performance of the predictor is considered. Our choice of feature selection with wrapper method is because it gives better results than filter methods, though the later is less in computational time than wrappers [15]. Therefore, in our proposed approach, we used the wrapper method to enhance the performance of NB classifier. Meanwhile, optimizing NB predictor has been used as part of feature selection with the wrapper method in order to reduce its computational time since the prediction of Naive Bayes is computationally less expensive than using other type of predictors [23].

B. Basic Procedure

We summarize our experimental procedure for the proposed approach to enhance the performance of Naive Bayes classier with discretizationas as follows:

- apply each of the four supervised and unsupervised discetization methods presented in Section II to NSL-KDD dataset subset. This leads to obtaining four discretized versions of the dataset subset.
- apply the wrapper feature selection method based on best first search method that involves optimizing Naive Bayes predictor as part of the selection to each of the four discretized versions of the dataset subset resulting in the previous step. Therefore, we obtain four subsets of selected features. Each one is resulting from applying feature selection to one discretized version of the dataset.
- apply each of the four subsets of selected feature obtained in the previous step to one of the four discretized version of the dataset subset obtained in step 1. Therefore we obtain 16 reduced discretized versions of each datset subset.
- test the performance of classifier with each of the 16 reduced discretized versions of each dataset subset obtained in the previous. step.

C. Performance evaluation

For each subset of NSL-KDD dataset, we have conducted one of the four discretization methods presented in Section II (EBD, EWBD, EFBD and PKID) before conducting feature selection with wrapper attribute evaluator (WrapperSubsetEval in Weka). Therefore, four combinations of discretizador plus WrapperSubsetEval were considered for each dataset subset. Table IV shows the subset of features that is selected by each combination.

We notice from the table that no two combinations exist that choose the same subset of features. Each combination leads to a different subset of features. Each of the four obtained subsets of features is applied to four discretized versions of the dataset subset(where each version is discretizated with one of the afro-mentioned discretization methods). Thus, in the end, we obtain 16 different reduced discretized version of each dataset subset. Next NB classifier is run on those versions of the dataset subset and hence different performance results of NB classifier could be obtained with each of those versions. Therefore, our approach for improving the performance of NB classifier with discretization aims to choose the best combination of discretizor, wrapper attribute evaluator and discretized version of the dataset. To achieve this, the discretized version of the

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE IV
SELECTED FEATURES OBTAINED BY A COMBINATIONS BETWEEN DISCRETIZORS AND WRAPPER ATTRIBUTE EVALUATOR.

Dataset	Combination	Selected features
NSL-KDD-10%	EBD-wrapper	3,5,8,10,13,16,24,33
	EWBD-wrapper	3,4,5,8
	EFBD-wrapper	3,5,6,8,13,23,26,32,38
	PKID-wrapper	3,5,10,13,15,17,26
NSL-KDD-20%	EBD-wrapper	3,5,6,10,16,23,25,33,36,38
	EWBD-wrapper	3,4,15,19
	EFBD-wrapper	2,3,5,10,13,16,24,32,38
	PKID-wrapper	1,3,5,10,13,30,36
NSL-KDD-30%	EBD-wrapper	3,5,6,7,10,16,19,22,23,26,30,36
	EWBD-wrapper	2,3,4,8,10,11,14,15,17,22,25,32,36,40
	EFBD-wrapper	3,5,8,10,13,16,17,19,22,37,38
	PKID-wrapper	1,3,5,7,8,10,13,30,36

dataset is chosen in such a way that is when it is reduced with the subset of selected features obtained by the combining the discretizor and wrapper attribute evaluator, it leads to the best performance of NB classifier that runs on this version.

For each NSL-KDD dataset subset, a total of 16 combinations of discretizator, wrapper attribute evaluator and discretized version of the dataset have been tested to obtain 16 reduced discretized versions of the dataset. Therefore, a total of 48 combinations were tested for the three experimental NSL-KDD dataset subsets. For the sake of clarity, only the combination that leads to the best accuracy of NB is shown. For example, for NSL-KDD 10%, the combination of EBD with WrapperSubsetEval results a subset of 11 features. When this subset of features is applied to EBD discretized version of the dataset before classification, we obtain a reduced discretized version of the dataset that leads to the best accuracy of NB of 98.32% .

Table V shows the performance results of N.B for the proposed approach. The second and third columns in the table indicate the combination of discretizator, wrapper attribute evaluator and discretized version of the dataset, and the corresponding number of selected features respectively that leads to the best accuracy of FN.

As shown in Table V, the combination of EFBD discretizator, wrapper attribute evaluator and EBD discretized version of the dataset appears to be the best combination for NSL-KDD-10% dataset; while the combination of EBD discretizator, wrapper attribute evaluator and PKID discretized version of the datset is the best combination for NSL-KDD-20% dataset. Finally, for NSL-KDD-30% dataset, the combination of EBD discretizator, wrapper attribute evaluator and EBD discretized version of the dataset is the best choice that leads to the best accuracy of NB. We also notice from the table that the performance of NB using our proposed approach outperforms the performance of SVM without discretization (as shown in Table I) and has become more closed to the performance of either SVM or RF with discretization.

VI. RELATED WORK

V. Bolon-Canedo et al. [15] have proposed an approach that combines a discretizator, a filter method for feature selection and a very simple classical classifier for enhancing the capability of classifier to identify network intrusions in KDD Cup 99 dataset. They used five methods of discretization including the supervised EBD, unsupervised PKID, the unsupervised EWBD, the unsupervised EFBD and Bin-log l (BL) discretization method in combination with three filters methods for feature selection implemented in Weka including the CFS, INTERACT and Consistency-based Filter to improve the performance of either NB or C4.5 classifiers.

H. Tribak et al. [16] proposed an intrusion detection system and evaluated it with the NSL-KDD intrusion detection dataset. They applied different learning algorithms on this dataset to identify attack connections from the normal ones and compared the performance of those algorithms in different scenarios including discretization, and features selections. They built two different discretized versions of NSL-KDD dataset in addition to the original one: the first is a dataset built by applying the supervised entropy-based discretization (EBD) and the second is a dataset built by the unsupervised equal-frequency binning discretization (EFBD). Next they train and test each learning algorithm on the original dataset, and its first and second versions. They found that the accuracy of their proposed intrusion detection system is improved by either supervised or unsupervised discretization techniques that they used.

M. Revathi and T.Ramesh [18] applied dimensionality reduction as a pre-processing step before classification to KDD Cup 99 dataset for the purpose of identifying the type of network attack. Their procedure for reducing dimensionality was based on performing data discretization, then applying feature selection to filter out redundant features. Finally, they used the selected features for performing classification. However, the type and method of discretization that they used in their procedure was not clear or mentioned explicitly.

Z. Shi [17] et al. proposed a supervised discretization algorithm based on information loss and an attribute reduction algorithm based on information gain. They used the former algorithm to process the KDD Cup 99 dataset. The discretized data is subjected to feature reduction by the later algorithm in order to reduce some unnecessary or redundant attributes. Next the remaining attributes are used to identify intrusions by an

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE V
THE BEST COMBINATIONS BETWEEN DISCRETIZORS, WRAPPER ATTRIBUTE EVALUATOR AND A DISCRETIZED VERSION OF THE DATASET THAT LEADS TO THE BEST ACCURACY OF NB.

Dataset	Combination	Selected features	NB Accuracy	NB FPR	NB FNR
NSL-KDD-10%	EFBD-wrapper-EBD	3,5,6,8,13,23,26,32,38	98.7%	0.024	0.004
NSL-KDD-20%	EBD-wrapper-PKID	3,5,6,10,16,23,25,33,36,38	99.03%	0.012	0.007
NSL-KDD-30%	EBD-wrapper-EBD	3,5,6,7,10,16,19,22,23,26,30,36	99.23%	0.013	0.003

intrusion detection system.

The work presented in this paper has been inspired by the one presented in [14]. It investigated the effect of supervised discretization methods on improving the classification performance on high dimensional biomedical datasets. Unlike this work, our work investigates the effect supervised versus unsupervised discretization methods on improving the performance of classifiers that run on high dimensional network intrusions datasets.

VII. CONCLUSIONS

In this paper, we have investigated the effect of preprocessing high dimensional network intrusions dataset with either supervised or unsupervised discretization techniques on improving the performance of classifiers for identifying network intrusions. We have applied several unsupervised and supervised discretization methods to NSL-KDD dataset, an improved version of the high dimensional KDD-cup dataset that has been widely used for network intrusion detection. Next, we have applied three supervised classification algorithms that are famous for their robustness to the curse of dimensionality problem including SVM, RF and NB to the obtained discretized versions of the original dataset. Our results show that applying discretization as a preprocessing step to classification does not lead to improve the performance of RF. However it leads to a little better improvement in the performance of SVM and a significant improvement in performance of NB, particularly using either the supervised EBD or the unsupervised PKID methods. However, the performance of NB with discretization has been found to be less than the performance of SVM and RF either with or without discretization. Finally, the paper presents an approach for improving the performance of NB with discretization. This approach relies on applying feature selection by a combination of discretization and wrapper method to a dataset that has been preprocessed by discretization then applying NB to the obtained reduced discretizated version of the dataset. The proposed approach makes the performance of NB outperforms the performance SVM without discretization and very closed to the performance of either SVM or RF with discretization.

As a future work, first, we are planning for investigating the effect of preprocessing the high dimensional datasets with either supervised or unsupervised techniques on improving the performance of other classification algorithms that are also known for their robustness to the curse of dimensionality problem such as neural networks, decision trees and rule-based classifiers. Second, we are also looking forward to testing the efficiency of our proposed approach when it combines discretization with wrapper feature selection that uses search techniques different from best first search, on improving the performance of classification with discretization.

VIII. ACKNOWLEDGMENT

The work presented in this paper has been neither published nor submitted for publication, in whole or in part, either in a serial, professional journal or as a part in a book which is formally published and made available to the public.

REFERENCES

[1] A. Lazarevic, J. Srivastava and V. Kumar. Data Mining for Intrusion Detection. *Tutorial on the Pacific-Asia Conference on Knowledge Discovery in Databases*, 2003.

[2] S. Kotsiantis, D.Kanellopoulos. Discretization Techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, Vol.32, No.1, pp. 47-58, 2006.

[3] G. Agre and S. Peev. On Supervised and Unsupervised Discretization. *Cybernetics and Information Technologies*, Vol.2, No.2, 2002.

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A Detailed Analysis of the KDD CUP 99 Data Set. *In Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, 2009.

[5] S. Revath and Dr. A. Malathi. A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2, Issue 12, December 2013.

[6] W. Wang and J. Yang. Mining High-dimensional Data. *Data Mining and Knowledge Discovery Handbook*, Chapter 27, pp. 793-799, 2005.

[7] J. Dougherty , R. Kohavi and M. Sahami. Supervised and Unsupervised Discretization of Continuous Features. *In Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, California, USA, July 9-12, 1995.

[8] R. Kohavi and M. Sahami. Error-Based and Entropy-Based Discretization of Continuous Features. *In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 114-119, 1996.

[9] M. Boulle. Khiops: A Statistical Discretization Method of Continuous Attributes. *Machine Learning Journal*, Vol. 5, Issue 1, pp 5369, April, 2004.

[10] U.Fayyad, K.Irani. Multi-interval Discretization of Continuous-Valued Attributes for Classification Learning. *In Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp.1022-1027, 1993.

[11] P. D. Grunwald. Introducing the Minimum Description Length Principle. *In Advances in minimum description length: theory and applications*, pp.322. MIT Press, Cambridge, MA, 2005.

[12] Y. Yang and G.I. Webb. Proportional k-Interval Discretization for Naive-Bayes Classifiers. *In Proceedings of the 12th European Conference on Machine Learning (EMCL 01)*, pp. 564-575, Springer-Verlag, 2001.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[13] Y. Yang and G.I. Webb. A Comparative Study of Discretization Methods for Naive-Bayes Classifiers. *In Proceedings of the 2002 Pacific Rim Knowledge Acquisition Workshop (PKAW 2002)*, pp. 159-173, Tokyo, Japan,2002.

[14] J. L. Lustgarten, V. Gopalakrishnan, H. Grover, S. Visweswaran. Improving Classification Performance with Discretization on Biomedical Datasets. *In Proceedings of American Medical Informatics Association Annual Symposium (AMIA 2008)*, Washington, DC, USA, November 8-12, 2008.

[15] V. Bolon-Canedo, N. Sanchez-Marono and A. Alonso-Betanzos. A combination of discretization and filter methods for improving classification performance in KDD Cup 99 dataset. *In Proceedings of International Joint Conference on Neural Networks*, Atlanta, Georgia, USA, June 14-19, 2009.

[16] H. Tribak, O. Valenzuela, F. Rojas and I. Rojas. Statistical Analysis of Different Artificial Intelligent Techniques applied to Intrusion Detection System. *In Proceedings of the 3rd International Conference on Multimedia Computing and Systems (ICMCS12)*, 2012.

[17] Z. Shi, Y. Xia, F. Wu and J. Dai. The Discretization Algorithm for Rough Data and Its Application to Intrusion Detection. *Journal of Network*, Vol. 9, No. 6, June 2014.

[18] M. Revathi and T.Ramesh. Network Intrusion Detection System Using Reduced Dimensionality. *Indian Journal of Computer Science and Engineering (IJCSE)*, Vol. 2, No. 1, 2011.

[19] Weka 3: Data Mining Software in Java. Available at: http://www.cs.waikato.ac.nz/ml/weka/.

[20] http://weka.wikispaces.com/Primer.

[21] R. Longadge, S. S. Dongre, L. Malik. Class Imbalance Problem in Data Mining: Review. *Class Imbalance Problem in Data Mining: Review*, Vol. 2, Issue 1, 2013.

[22] L. T. Heberlein. Statistical Problems with Statistical based Intrusion Detection. *Technical report*, Version1, Net Squared, Inc., 2007.

[23] K. Ming Leung. Naive Bayesian Classifier. Department of Computer Science / Finance and Risk Engineering, Polytechnic University, 2007.

[24] R. Kohavi and G. H.John. Wrappers for feature subset selection. *Artificial Intelligence*,Vol.97, Issue 1-2, pp. 273-324, 1997.

[25] C. Mihaescu. Naive-Bayes Classification Algorithm Laboratory module 4. Avialable at: http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf.

[26] J. Weston. Support Vector Machine (and Statistical Learning Theory) Tutorial. NEC Labs America.

[27] G.Louppe. Understanding Random Forests from Theory to practice. PhD dissertation, Faculty of Applied Sciences, University of Lige, 2014.

[28] H. Liu and R. Setiono. Feature Selection via Discretization. *IEEE Transactions on Knowledge and Data Engineering*, VOL. 9, NO. 4, July/Augst, 1997.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500