Network Intrusion Detection Using a Hidden Naïve Bayes Binary Classifier

Levent Koc and Alan D. Carswell
Center for Security Studies,
University of Maryland University College,
Adelphi, Maryland, USA

***Abstract* — Using data mining techniques in intrusion detection systems is common for the classification of the network events as either normal events or attack events. Naïve Bayes (NB) method is a simple, efficient and popular data mining method that is built on conditional independence of attributes assumption. Hidden Naïve Bayes (HNB) is an extended form of NB that keeps the NB's simplicity and efficiency while relaxing its independence assumption. Our experimental research claims that the HNB binary classifier model can be applied to intrusion detection problem. Experiment results using classic KDD 1999 Cup intrusion detection dataset indicate that HNB binary classifier has better performance in terms of detection accuracy compared to the traditional NB classifier.**

Keywords- intrusion detection systems, data mining, machine learning, classifiers, Naïve Bayes, Hidden Naïve Bayes (HNB)

I. INTRODUCTION

With the increasing expectation on detection performance in terms of accuracy and speed in spite of the dynamic network environments, vast volumes of network data, and limited processing power, network intrusion detection is still an important and challenging problem [1]. Classification is one of the data mining techniques that are often utilized to differentiate malicious events and normal events out of all network events. It identifies the class (category) labels (e.g. malicious vs. normal) of network event records (instances) which consist of features (attributes). Learning classifier models rely on the premise that classifier model is built by learning from the provided training data, and then the class labels for the instances of the new data are predicted using the built classifier model. Numerous classifier models have been applied to the intrusion detection problem, including rule-based detection [2], neural networks [3-5], fuzzy logic [6], Bayesian analysis [7], Naïve Bayes [8] and the hidden Markov model [9].

Our research study proposes an intrusion detection model based on a binary classifier that is used to classify the network events as normal or attack events and it is built on a new data mining method called Hidden Naïve Bayes (HNB) [10,11]. HNB method is an extended form of Naïve Bayes (NB) method which is a simple and efficient data mining method that is built on conditional independence of attributes assumption. In recent research studies, the HNB multiclass classifier model is applied to network intrusion detection and shows encouraging results compared to the traditional NB [12]. Our experimental simulation study explores the performance improvements with the application of the HNB binary classifier model to the intrusion detection problem. It claims that the performance of the binary classifier based on the HNB method is better than the one based on the traditional NB method in terms of detection accuracy for the intrusion detection problem. We modelled and tested our claim using the classic Knowledge Discovery and Data Mining Cup 1999 (KDD'99) [13] intrusion detection dataset.

In the Related Work section, we present background information for both traditional NB method and one of its extended forms, HNB method. Our intrusion detection model and research framework are introduced in the Research Method section.

Our experiment setup and results are presented in the Experiments and Results section. The results obtained with the HNB binary classifier against those obtained with the traditional NB classifier are compared and discussed in the same section. We conclude our article in the last section with a brief summary of the findings and implications of our research study.

II. RELATED WORK

A classifier function maps each instance of a dataset to a distinct class by prediction. More specifically in the intrusion detection domain, a binary classifier maps the network events to either a normal event class or an attack event class, while a multiclass classifier further maps the attack event class to denial of service (DOS), probe, remote to local user (R2L) and user to root (U2R) attack classes.

A. Naïve Bayes Classifiers

NB classifier model is the simplest form of Bayesian Network classifier. Its main premise derives from its simplicity which relies on the independence of attributes assumption.

A Bayesian classifier maps the feature set of *A* which consists of $\{a_1, a_2, .., a_n\}$ into class set of *C* that consists of $\{c_1, c_2, .., c_n\}$ on a dataset *D* which consists of $\{E_1, E_2, .., E_t\}$ instances and can be defined as the equation (1):

$$c(E) = \arg \max_{c \in C} P(c)P(a_1, a_2, \ldots, a_n|c). \quad (1)$$

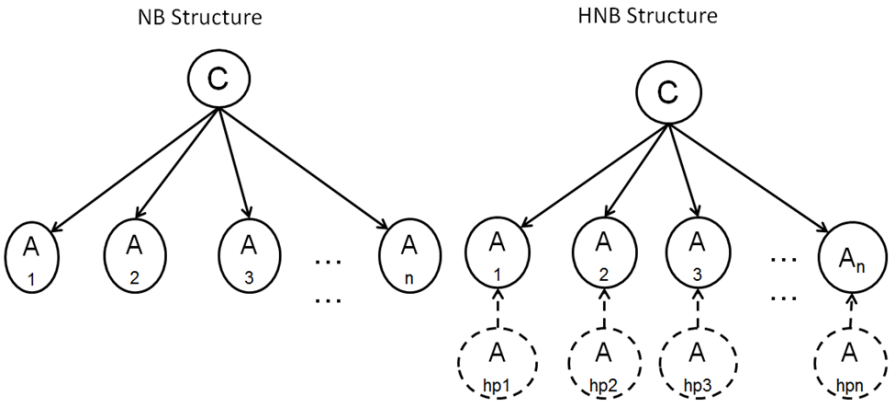

Figure 1 NB and HNB Structures

Then, with the consideration of naïve assumption of the independence of the attributes given the class as in (2), NB classifier is illustrated in Figure 1 and defined in (3).

$$P(E|c) = P(a_1, a_2, \ldots, a_n|c) = \prod_{i=1}^{n} P(a_i|c). \tag{2}$$

With the conditional independence of attributes assumption, it is simple to compute $P(C)$ and $P(a_i|c)$, and construct a NB classifier. In addition to its simplicity and efficiency, NB classifier provides comparable accuracy performance which is similar to performance of other leading classification methods including neural networks and classification trees [14].

$$c(E) = arg \max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|c). \tag{3}$$

NB classifier performs well when it is used on datasets which its conditional independence assumption property is accurately satisfied. However, it does not perform well when the independence assumption property is not satisfied in the datasets [11] that have complex attribute dependencies as in the KDD'99 intrusion detection dataset. It is also reported that the NB classifier's accuracy performance declines in the large datasets [15].

One of the earlier applications of the Bayesian classifier model in intrusion detection area is Barbara study [7] which is called Audit Data Analysis and Mining (ADAM). ADAM is based on an anomaly detection system built upon pseudo-Bayes estimators to estimate the prior and posterior probabilities of new attacks. These probabilities are used to construct a NB classifier for classifying normal and attack events without prior knowledge about new attacks.

B. Hidden Naïve Bayes Classifiers

In the last two decades, there were many studies focused on reducing the independence assumption of NB models including the tree-augmented naïve Bayes (TAN) [16], averaged one-dependence estimators (AODE) [17] and hidden naive Bayes (HNB) [10]. As an extended form of the NB classifier, HNB classifier is based on the creation of an additional layer that represents a hidden parent of each attribute as shown in Figure 1. The purpose of the hidden parent (A_{hpi}) is to combine the weighted influences from all of the other attributes (A_i) [10,11]. The joint distribution is defined as

$$P(A_1, \ldots, A_n|C) = P(C) \prod_{i=1}^{n} P(A_i|A_{hp_i}, C). \tag{4}$$

where

$$P(A_i|A_{hp_i}, C) = P(C) \sum_{j=1, j \neq i}^{n} W_{ij} * P(A_i|A_j, C). \tag{5}$$

The HNB classifier can be defined as

$$c(E) = arg \max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|a_{hp_i}, c). \tag{6}$$

where

$$P(a_i|a_{hp_i}, c) = P(c) \sum_{j=1, j \neq i}^{n} W_{ij} * P(a_i|a_j, c). \tag{7}$$

One way to calculate the weights W_{ij}, where $i,j = 1, 2,\ldots, n$ and i is not equal to j, is to use the conditional mutual information (CMI) between two attributes A_i and A_j as the weight of $P(A_i|A_j, C)$, as shown in (8) [10].

$$W_{ij} = \frac{I_p(A_i; A_j|C)}{\sum_{j=1, j \neq i}^{n} I_p(A_i; A_j|C)} \tag{8}$$

TABLE I. PROPERTIES OF THE KDD'99 DATASET (BINARY)

Class	Number of Records in Training Data	Distributions of Training Data	Number of Records in Test Data	Distributions of Test Data
Normal	97278	19.69%	60593	19.48%
Attack	396743	80.31%	250436	80.52%
Total	494021	100.00%	311029	100.00%

where $Ip(A_i,A_j|C)$ is the CMI defined in (9).

$$I_p(A_i;A_j|C) = \sum_{a_i,a_j,c} P(a_i,a_j,c) log \frac{P(a_i,a_j|c)}{P(a_i|c)P(a_j|c)} \quad (9)$$

A recent study [12] applied HNB based multiclass classifier to the intrusion detection domain with promising results in terms of detection accuracy.

III. RESEARCH METHOD

With the anticipation of the good performance of the HNB multiclass classifier from earlier studies [10,12], this study explores the application of HNB method as a binary classifier to intrusion detection problem. The results achieved with the HNB binary classifier are compared with those obtained with the traditional NB classifier.

We used the data sample that was developed by the MIT Lincoln Laboratory as part of the 1998 DARPA intrusion detection evaluation offline dataset [13]. While we note the reported limitations [18], the KDD'99 dataset with its interesting properties is considered as a classic challenge in intrusion detection research domain. This dataset is used in our experiments, because it is the most comprehensive public dataset available and it is still commonly used to evaluate and analyze the performance in the intrusion detection studies.

The dataset contains training and test data that include network traffic connection records in the form of TCP dump data. We used specifically the 10% KDD'99 dataset which contains the labeled records.

One of the unique properties of this dataset is that the number of the records and class distributions are not same in the training and test datasets. As shown in Table I, about 20% of the records are labeled as normal connection events. Each connection record contains 34 continuous, 7 discrete and total 41 features. Since the NB binary classifier model is built using discrete attributes, continuous attributes are required to be converted into discrete counterparts using a discretization method. It is reported that the discretization also helps the performance improvements of classifier models on large datasets [19], including the KDD'99 dataset [20]. In our study, we used the entropy minimization discretization (EMD) method because of its performance benefits for the NB classifier method on the KDD'99 dataset [20].

EMD method is based on the minimum entropy heuristic required to discretize continuous attributes. It picks a cut point for discretization based on the class entropy of the candidate partitions; this cut point is then recursively applied to the created intervals until the stopping condition is reached [21]. This stopping condition is found based on the minimum description length (MDL) algorithm.

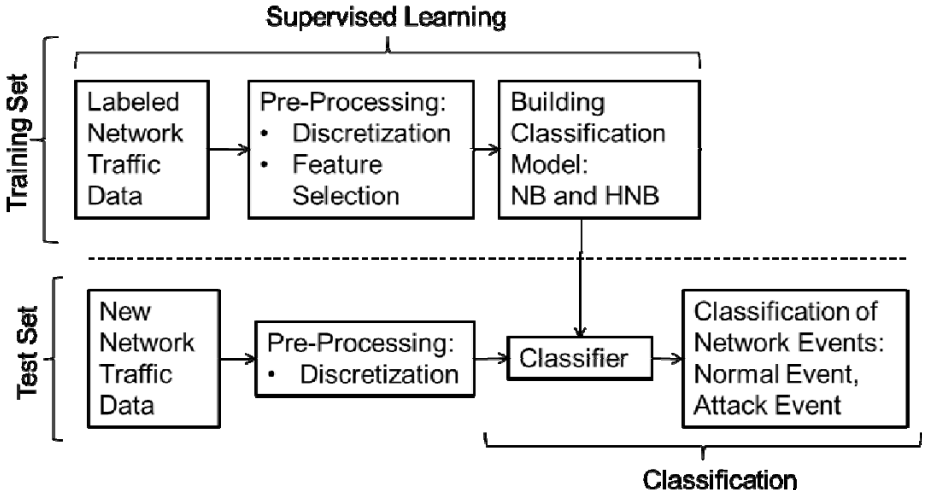

Figure 2 Research Framework

Due to large feature set of a dataset, high data dimensionality and possible interdependence among some features create significant challenges to any data mining model. As in many other data mining techniques, selecting the right features and reducing the number of features is an important task for the efficient processing speed and reducing the irrelevant, redundant and noisy data for predictive accuracy. As illustrated in our research framework on Figure 2, feature selection step in pre-processing phase is one of the common approaches to these challenges.

The feature selection method applied is consistency-based (CONS) feature selection filter method which performs well on the KDD'99 dataset using the NB classifier method [20].

CONS method [22,23] built on an inconsistency criterion that specifies the extent to which the dimensionally reduced data can be accepted. In each round, it generates a random subset and more consistent set is recorded.

We used accuracy, error rate and area under receiver operating characteristics (ROC) curve as performance evaluation methods in this study. These methods are commonly accepted to summarize and compare the classifier performance [24]. Accuracy is the fraction of the correctly classified instances. Error rate is the fraction of the misclassified instances in a dataset.

As illustrated in Table II, a confusion matrix provides a summary of the performance of a binary classifier algorithm. While normal and attack columns represent the predicted class, normal and attack rows represent the actual class. The intersection of the columns and rows provides corresponding counts which allows basic summary of the performance of the model algorithm. While false positive (FP) count gives false alarms in the model, true positive (TP) count gives the correctly detected attacks.

TABLE II. CONFUSION MATRIX FOR A BINARY CLASSIFIER

Confusion Matrix		**Predicted Class**	
		Normal	**Attack**
Actual Class	**Normal**	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

In terms of counts illustrated in a confusion matrix, accuracy and error rate can be defined as

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

and

$$Error\ Rate = 1 - Accuracy \quad (11)$$

A ROC graph illustrates relative tradeoffs on the TP and FP results. It is generally used for cost benefit analysis. A method is used to reduce the representation of the ROC graph into a singular scalar value to measure and compare classifier performance. This method relies on the calculation of the area under the ROC curve (AUC) [25]. It is generally accepted that the closer the AUC value to 1, higher the classifier's performance in terms of accuracy.

IV. Experiments and Results

Using the research framework given in Figure 2, we executed our simulation experiments using Weka tool, which is an open source machine learning software written in java [26].

As the first step in pre-processing phase of our framework dictates, we applied supervised *discretize()* method for EMD discretization of the continuous variables of the 10% KDD'99 dataset. We proceeded with the feature selection step by applying the consistency subset filtering with the default values. Seven attributes *duration, service, src_bytes, count, dst_host_srv_count, dst_host_same_srv_rate* and *dst_host_diff_srv_rate* were selected.

After discretization and feature selection steps, we build our binary classifier models for NB and HNB using the pre-processed dataset. To build the classifier models, we applied 10-fold cross-validation method on training dataset as part of the supervised learning phase. 10-fold cross-validation is a popular method which generally works well with the sufficient datasets. The method relies on the idea that dataset divided randomly into 10 disjoint subsets of approximately equal size. In each run, out of 10 runs, one of these subsets is used as the test set and the remaining 9 sets are training set to build a classifier model. Mean of the estimates for each run provides an accuracy estimate for building the classifier [24,26].

TABLE III. CONFUSION MATRIX FOR THE NB BINARY CLASSIFIER

Confusion Matrix		**Predicted Class**	
		Normal	**Attack**
Actual Class	**Normal**	59817	776
	Attack	26327	224109

TABLE IV. CONFUSION MATRIX FOR THE HNB BINARY CLASSIFIER

Confusion Matrix		**Predicted Class**	
		Normal	**Attack**
Actual Class	**Normal**	59520	1073
	Attack	19463	230973

We then applied the classifier models built from the training dataset to the test dataset. The confusion matrices

that are provided in the Table III and IV summarize the performance of the classifier models in terms of instance counts correctly or incorrectly predicted. As shown in these two confusion matrix tables, the number of correctly detected attack by HNB binary classifier model is much higher than the one by NB binary classifier model.

TABLE V. TEST RESULTS FOR THE CLASSIFIER PERFORMANCE

Model	Accuracy	Error Rate	*AUC**
NB	0.9129	0.0871	0.9400
HNB	0.9340	0.0660	0.9790

* Weighted Average

The experiment results based on the test dataset in terms of accuracy, error rate and AUC are provided in Table V. These results specify that HNB binary classifier based intrusion detection model's accuracy is 0.9340, error rate is 0.0660, and the AUC value is 0.979. Results also indicate traditional NB based model's accuracy is 0.9129, error rate is 0.0871, and the AUC value is 0.9400.

Figure 3 and 4 also illustrate that HNB binary classifier based model always closer to the top left corner and occupy a larger AUC than the NB based model. These findings indicate that HNB binary classifier based intrusion detection model performs better than NB based model in terms of all three measures, accuracy, error rate and AUC.

These results are consistent with earlier studies on the HNB where significant performance improvements of HNB were demonstrated against traditional NB method using other datasets [10] and KDD'99 dataset [12].

V. CONCLUSION

In this article, we explained the increasing need to apply data mining methods to classify network attack events. We reviewed a simple and widely used data mining method which is called Naïve Bayes (NB) classifier model that utilizes on the independence of attributes assumption. We introduced a binary classifier model based on the Hidden Naïve Bayes (HNB) method as an extension to NB to reduce its naivety assumption while keeping its simplicity and efficiency.

We applied this new classifier method to the challenging network intrusion detection problem and tested performance of our model using the well-recognized KDD'99 intrusion detection dataset. Our experimental study outcomes indicate that the HNB binary classification model augmented with EMD discretization and CONS feature selection filter methods has better overall results in terms of detection accuracy, error rate and area under ROC curve than the traditional NB model.

With its simplicity inherited from the traditional NB model and its advantage over the NB model's conditional independence assumption, HNB binary classifier is a promising model for datasets with dependent attributes, such as the KDD'99 intrusion detection dataset.

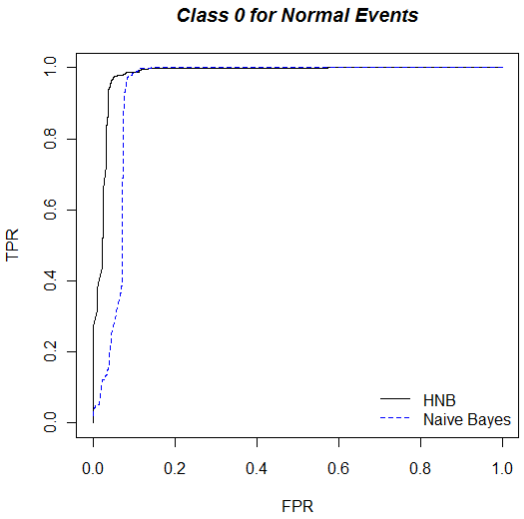

Figure 3 ROC graph for detection of normal events

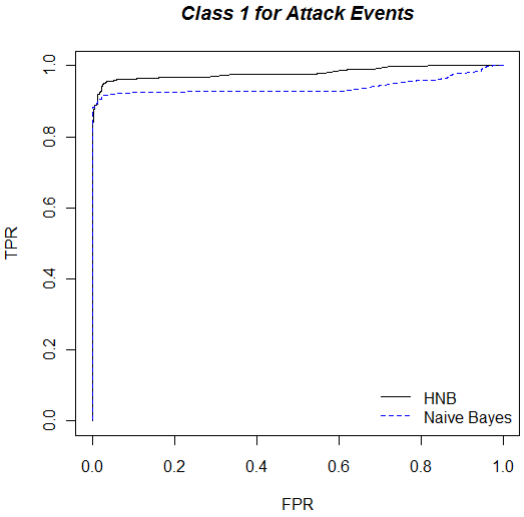

Figure 4 ROC graph for detection of attack events

REFERENCES

[1] P. Kabiri and A. A. Ghorbani, "Research on intrusion detection and response: A survey," International Journal of Network Security, vol. 1, pp. 84-102, 2005.

[2] T. F. Lunt, "Real-time intrusion detection," in COMPCON Spring '89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers., 1989, pp. 348-353.

[3] J. Cannady, "The application of artificial neural networks to misuse detection: initial results," in Proceedings of the Recent Advances in Intrusion Detection '98 Conference, Louvain-la-Neuve, Belgium, 1998, pp. 31-47.

[4] R. P. Lippmann, Cunningham, R. K., "Improving intrusion detection performance using keyword selection and neural networks," Computer Networks, vol. 34, pp. 597-603, 2000.

[5] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in Proceedings of the 2001 IEEE workshop on information assurance and security, United States Military Academy, West Point, New York, 2001, pp. 85-90.

[6] S. M. Bridges, Vaughn, R. B., "Fuzzy data mining and genetic algorithms applied to intrusion detection," in Proceedings of the 23rd National Information Systems Security Conference (NISSC), 2000, Baltimore, Maryland, 2000.

[7] D. Barbara, Wu, N., Jajodia, S., "Detecting novel network intrusions using bayes estimators," in First SIAM Conference on Data Mining, Chicago, IL, 2001.

[8] M. Panda and M. R. Patra, "Network intrusion detection using naive Bayes," International Journal of Computer Science and Network Security, vol. 7, pp. 258-263, 2007.

[9] B. Gao, HY. Ma, and YH. Yang "HMMs (Hidden Markov Models) based on anomaly intrusion detection method," in International Conference on Machine Learning and Cybernetics, 2002, 2002, pp. 381-385.

[10] L. Jiang, Z. Harry, and C. Zhihua, "A Novel Bayes Model: Hidden Naive Bayes," Knowledge and Data Engineering, IEEE Transactions on, vol. 21, pp. 1361-1371, 2009.

[11] J. Yaguang, Songnian, Y., Yafeng, Z., "A novel Naive Bayes model: Packaged Hidden Naive Bayes," in Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International, 2011, pp. 484-487.

[12] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier," Expert Systems with Applications, vol. 39, pp. 13492-13500, 2012.

[13] KDD-Cup. (1999, July 29, 2011). KDD Cup 1999 Data. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[14] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in Proceedings of the tenth national conference on artificial intelligence, San Jose, California, 1992, pp. 223-228.

[15] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid," in Proceedings of Second International Conference Knowledge Discovery and Data Mining (KDD'96), Portland, Oregon, 1996, pp. 202-207.

[16] N. Friedman, D. Geiger, and M. Goldszmidt, (1997). "Bayesian network classifiers". Machine learning, 29(2-3), 131-163.

[17] G. I. Webb, R. J. Boughton, and Z. Wang. "Not so naive Bayes: aggregating one-dependence estimators." Machine learning 58.1 (2005): 5-24.

[18] M. Tavallaee, Bagheri, E., Lu, W., Ghorbani, A.A., "A detailed analysis of the kdd cup 99 data set," in IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009), Ottawa, Canada, 2009, pp. 53-58.

[19] H. Liu, Hussain, F., Tan, C. L., Dash, M., "Discretization: an enabling technique," Data Mining and Knowledge Discovery, vol. 6, pp. 393-423, 2002.

[20] V. Bolon-Canedo, Sanchez-Maroo, N., Alonso-Betanzos, A., "A combination of discretization and filter methods for improving classification performance in KDD Cup 99 dataset," in International Joint Conference on Neural Networks, IJCNN 2009, Atlanta, Georgia, 2009, pp. 359-366.

[21] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in Proceedings of the 13th International Joint Conference on Artifical Itelligence, IJCAI'93, Chambéry, France, 1993, pp. 1022-1029.

[22] M. Dash and H. Liu, "Consistency-based search in feature selection," Artificial Intelligence, vol. 151, pp. 155-176, 2003.

[23] L. Huan and R. Setiono, "Feature selection via discretization," Knowledge and Data Engineering, IEEE Transactions on, vol. 9, pp. 642-645, 1997.

[24] N. Japkowicz, Shah, M., Evaluating Learning Algorithms: a classification perspective. New York: Cambridge University Press, 2011.

[25] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, pp. 861-874, 2006.

[26] I. H. Witten, E. Frank, M. A. Hall, Data mining : practical machine learning tools and techniques, 3rd ed. Burlington, MA: Morgan Kaufmann, 2011.