Network Intrusion Detection with Cascading Classification

Stanislav Marcek Martin Drozda
Slovak University of Technology, Faculty of Electrical Engineering and Information Technology,
Ilkovičova 3, 81219 Bratislava, Slovakia.
Email: stanislav.marcek@stuba.sk, martin.drozda@stuba.sk

***Abstract*—The KDD99 network intrusion contest and the related intrusion data sets attracted increased attention of the research community. The success rate of contest participants was evaluated in terms of the obtained classification cost. The classification cost of the contest winner was 0.2331, the best approach prior to our work carries the classification cost of 0.2224. We show that a simple approach based on cascading classification leads to the classification cost of 0.2079.**

Cascading classification is in our case done by applying 2-nearest-neighbor classification. The samples which could not be predicted with 2-nearest-neighbor classification (4-6%) are further classified with a clustering approach with class priority. This clustering approach when applied in isolation underperforms other approaches. However, when applied in cascading classification, it can take advantage of the reduced number of samples. We argue that cascading classification is a viable alternative in scenarios where less complex machine learning approaches are favorable, for example due to possible performance degradation in resource constrained devices such as mobile phones, embedded systems or sensors.

Keywords*-Exact Euclidean Locality Sensitive Hashing (E2LSH); k*-nearest-neighbor classification; cascading classification; KDD99 classifier learning contest; intrusion detection.

I. Introduction

Cascading classification is based on the rationale that a large number of samples in a data set can be correctly classified with a less complex classifier. At the next stage, a classifier, which targets a specific property of the remaining samples, can be applied. The number of stages in cascading classification reflects the number of such specific properties.

This form of multi-stage classification can also lead to cost efficiency. This can be achieved when classifiers are applied in increasing order of cost. The highest cost classifier is thus only applied to a sub-set of samples. This can be of advantage in distributed environments such as wireless mesh networks or sensor networks, where classification accuracy depends on information which needs to be transmitted over wireless medium. A more frequent information exchange can lead to a higher classification accuracy, however, each information exchange incurs a communication cost. This can give rise to a trade-off between classification accuracy and cost [1]. Cascading classification can thus be applied in scenarios where a specific trade-off needs to be achieved in order to keep costs in limits.

Table I
The KDD'99 cost matrix [2].

		Predicted				
		normal	probe	DOS	U2R	R2L
Actual	normal	0	1	2	2	2
	probe	1	0	2	2	2
	DOS	2	1	0	2	2
	U2R	3	2	2	0	2
	R2L	4	2	2	2	0

Our focus herein is on cascading classification applied to network intrusion detection. Since creating a data set which captures a network operation under normal behavior and under various attacks may require a long period of time, we rely on a benchmark data set. This data set was given to the participants of the classifier learning contest which was held during the 1999 Conference on Knowledge Discovery and Data Mining (KDD99).

The task for the participants of the KDD99 classifier learning contest was to correctly predict when network behavior is normal and when an attack is underway [2]. The contest participants were given a *training data set* well before the conference started. During the conference, the participants were presented with a *test data set* which had a different class distribution. It also included several novel attacks not present in the learning data set. The attacks fall into one of four different classes:

- probe - surveillance and other probing,
- denial of service (DOS),
- user-to-root (U2R) - unauthorized access with superuser (root) privileges
- remote-to-local (R2L) - unauthorized access from a remote machine.

The success rate of each participant was evaluated in terms of *classification cost*. After samples in the test data set were predicted, each contest participant submitted his/her confusion matrix for evaluation. A confusion matrix contains information about actual and predicted classifications done by a classifier. Then the classification cost $\xi \in R$ for each confusion matrix was computed. The average cost is defined as a sum of the products of confusion matrix and cost matrix (see Table I) divided by the number of total predicted

Table II
THE RESULT OF KDD99 WINNER [2]; $\xi = 0.2331$.

		Predicted				
		normal	probe	DOS	U2R	R2L
Actual	normal	60262	243	78	4	6
	probe	511	3471	184	0	0
	DOS	5299	1328	223226	0	0
	U2R	168	20	0	30	10
	R2L	14527	294	0	8	1360

samples:

$$\xi = \frac{\sum\limits_{i,j} c_{i,j} b_{i,j}}{\sum\limits_{i,j} c_{i,j}} \tag{1}$$

where $c_{i,j}$ is a count of samples known to be in class i but predicted to be in class j; $b_{i,j}$ is the corresponding entry of the cost matrix; $0 \leq i, j \leq 4$ (there were 4 intrusion classes and one normal class).

Each sample in either the learning or test data set had 41 features. The test data set contained the following numbers of samples: normal 60,593 (19.48%), probe 4,166 (1.34%), DOS 229,853 (73.9%), U2R 228 (0.07%) and R2L 16,189 (5.2%), together 311,029 labeled samples.

The rest of this document is organized as follows. In Section II we discuss the related work. In Section III we argue that 2-nearest-neighbor classification can lead to a more favorable classification cost than the complex approach applied by the KDD99 contest winner. We introduce cascading classification and discuss how it can take advantage of 2-nearest-neighbor classification. In Section IV we define the measures that we apply when reasoning about classifier performance. In Section V we present our results obtained by applying cascading classification, and finally, in Section VI we conclude.

II. Related Work

Since its introduction, the KDD intrusion detection data set attracted a lot of attention from the research community. An interesting fact remains that despite a continued research, the current best results with respect to classification cost are near the original contest winner, who, compared to his followers, had an incomparably shorter time to come up with an effective classification approach.

Table II shows the confusion matrix of the KDD99 competition winner [3]. The winning method was based on a set of C5 decision trees and a classification error minimization. The classification cost of the winning entry was 0.2331.

The 10th placed entry in the KDD99 contest was based on a 1-nearest-neighbor classifier [2]. The confusion matrix is shown in Table III. The classification cost was 0.2523. According to [2], any result with the classification cost of under 0.27 was considered by the contest organizers as well performing. Any result with the classification cost of over 0.29 was considered as inferior.

Table III
THE RESULT OF THE BEST 1-NEAREST-NEIGHBOR CLASSIFIER OF KDD99 [2]; $\xi = 0.2523$.

		Predicted				
		normal	probe	DOS	U2R	R2L
Actual	normal	60322	212	57	1	1
	probe	697	3125	342	0	2
	DOS	6144	76	223633	0	0
	U2R	209	5	1	8	5
	R2L	15785	308	1	0	95

Table IV
2-NEAREST-NEIGHBOR CLASSIFIER [4]; $\xi = 0.1767$ (NOT CONSIDERING UNDECIDED SAMPLES).

		Predicted					
		normal	probe	DOS	U2R	R2L	undecided
Actual	normal	58400	75	32	4	5	2077
	probe	10	2267	3	0	0	1886
	DOS	5174	0	221346	0	0	3333
	U2R	17	0	0	1	1	209
	R2L	10531	0	2	9	538	5109

Koc et al. [5] introduced a classification approach based on several types of classifiers. Their focus was on naive Bayes classification. As input they only applied discrete values which were obtained with several pre-processing discretization methods (Entropy Minimization Discretization, Proportional k-Interval Discretization). To lower computation cost they applied three filter feature selection methods (Correlation, Consistency, INTERACT); for a comparison see Sánchez-Maroño et al. [6]. The classification cost that they could obtain was 0.2224. This is to our best knowledge, approach offering the lowest classification cost. The authors do not provide any confusion matrix.

Xiang et al. describe hybrid classifier approach [7]. The critical step in their approach was to separate out the U2R, R2L and normal samples. They used unsupervised learning such as Bayesian and k-means clustering. The final set had 178 clusters. Then these clusters got labeled applying the rule that samples get predicted as normal samples only if none of the training samples within the clusters are U2R or R2L attacks. They did not compute the classification cost, however, they obtained a high detection rate for U2R (71.43%) and R2L (46.97%). Obtaining a high detection rate for these two classes is critical for obtaining a low classification cost. Note the high cost penalty in Table I for misclassifying R2L as normal behavior.

III. Can We Do Better?

Marcek et al. [4] proposed an approach which is based on approximate 2-nearest-neighbor computation. If both of the two approximate nearest neighbors shared the same label, then the sample would be predicted to belong to this class. If this is not the case, the test sample is marked as *undecided*.

The data sets for this approach were normalized and the

2-nearest-neighbor computation only considered neighbors within a certain distance R. If two nearest neighbors could not be found within R, then the prediction would be undecided as well. Table IV shows the results using 2-neighbor classification with R set to 0.1. The classification cost not considering undecided test samples was 0.1767. The rate of undecided samples was 4.06%.

There are several alternatives for dealing with undecided samples. One obvious alternative is to run a more complex classification algorithm on undecided samples in the hope that these additional classification will provide good results.

Let us now analyse the classification cost ξ when undecided samples get assigned to a given class. Note that the cost matrix shown in Table I was available to the contest participants. By assigning the undecided samples to a chosen class, the following classification cost is possible:

- $\xi = 0.2211$, if the undecided samples get assigned to the probe class. The rational is that assigning samples to this class carries the least cost; see Table I.
- $\xi = 0.2178$, if the undecided samples get assigned to the R2L class. This is the least classification cost possible with this approach.
- $\xi = 0.2647$, if the undecided samples get assigned to the normal class. This is worse than the best current approach, however, it offers a low false alarm rate of 0.052%.

If we assume that the undecided samples can be classified with an infallible classifier then we get the following cost:

- $\xi = 0.1695$, if each undecided sample gets correctly classified.

The *open question* that we address herein is whether a classification cost lower than 0.2178 is possible and how close we can get to 0.1695. Our approach is to apply a machine learning approach to classify undecided samples.

Let us now explain our approach in formal terms. Let K_1 and K_2 be classifiers. We first apply K_1 to classify samples, in our case K_1 is done as 2-nearest-neighbor classification. The samples that K_1 could not classify are sent to K_2 for further classification. We are thus looking at cascading classification with two classifiers. This can be generalized to a larger number of classifiers. Herein we use the following notation for two-classifier cascading classification:

$$K_1 \odot K_2.$$

Cascading (multi-stage) classification, where the next classifier in a classifier chain is applied upon the result of the previous classifier, was introduced by Kaynak and Alpaydin [8]. Therein the authors state *”at the next stage, using a costlier classifier, we build a more complex rule to cover those uncovered patterns of the previous stage”*.

Unlike other ensemble classification approaches such as stacking, boosting or bagging, cascading classification received much less attention from the research community. Gama and Brazdil [9] did an empirical study that applied several combinations of classifiers such as Bayes classifier, C4.5 decision tree classifier and linear discriminant function. They tested cascading classification on several standard data-sets from the UCI Repository [10]. Their results were non-conclusive about whether cascading classification can offer an advantage over single classifier approaches or other ensemble classification approaches.

Next, we define several key performance measures that we apply when evaluating classification performance.

IV. Classifier Performance Evaluation

Detection rate for class i is defined as:

$$DetR_i = \frac{c_{i,i}}{\sum\limits_j c_{j,i}} \tag{2}$$

False negatives rate can then be defined as:

$$FN_i = 1 - DetR_i \tag{3}$$

False positives rate is defined as follows:

$$FP_i = \frac{\sum\limits_j c_{i,j} - c_{i,i}}{\sum\limits_{k,j} c_{k,j} - \sum\limits_k c_{k,i}} \tag{4}$$

where $0 \leq k \leq 4$. Accuracy is the probability that a sample gets correctly classified:

$$Acc = \frac{\sum\limits_i c_{i,i}}{\sum\limits_{i,j} c_{i,j}} \tag{5}$$

V. Cascading Classification: Results

Two types of training data sets are available: complete and short. The complete training data set has about 5 million samples, whereas the shorter version has 494,021 samples. Both these training data sets include duplicate samples. We removed these duplicate samples and obtained two training data sets with 1,074,974 and 145,584 samples, respectively.

In our experiments, we used both these training data sets. When the shorter version is used, we indicate this fact with an asterisk *.

A. K_1 *Classification*

As previously discussed, K_1 is based on 2-nearest-neighbor classification. We apply an approximate approach to nearest neighbor classification due to Andoni and Indyk [11]. We use the implementation of approximate nearest-neighbor classification (E2LSH) from the same authors [12]. E2LSH considers samples within a certain distance R. In our experiments, we set this parameter to 0.1. E2LSH applies hashing to buckets as an underlying data structure. Since E2LSH is an approximate approach there is a certain chance of collision in a bucket. We set the collision parameter to 90%, which corresponds to 10% probability that nearest neighbor is not reported.

The advantage of this approximate nearest-neighbor classification is that it can be applied also in the case when the number of features is prohibitively large, for example there is support in the literature that decision tree classification does not scale well with the number of features [11]. Applying such an approximate method could be of advantage when cascading classification is applied to resource constrained devices such as mobile phones.

B. K_2 Classification

K_2 in our case evaluates undecided samples, which are a result of 2-nearest-neighbor classification. K_2 is implemented in Rapidminer tool [13], which also includes Weka [14], a machine learning library which implements a wide range of classification approaches.

The KDD99 data set contains 7 nominal features and 34 continuous features. Out of 7 nominal features 4 could be directly mapped to either 0 or 1. In order to allow for classification with E2LSH or X-Means clustering, the remaining nominal features have to be transformed.

We count unique values in each nominal feature. Then we map them into a point of dimensionality D, where D is number of unique values found in data set. In other words, the dimensionality of data set grows by D per feature mapping. For example $(1, 0, 0, 0, 0)$ corresponds to a unique value mapped to a point. This is repeated for each nominal feature that could not be directly mapped, i.e. in our case to 3 features.

The above procedure is applied to the training data set. The test data set includes features with unique values not mapped previously. Since the distance between two mapped unique values in training data set is constant (as per above described procedure), this rule is transferred to test set for un-mapped features. The rationale is that dimensionality of various data sets must remain unchanged.

For example, in a space of dimensionality 2, we are able to find 3 points which satisfy the condition of constant distance between them. That is, in a D-dimensional space, we are able to distribute $D+1$ points having constant mutual distance.

In order to obtain a good classification performance, we first apply forward feature selection. The goal of feature selection is to obtain a subset of features which are relevant in classification. Having less features leads to lower learning complexity, and thus to more efficient classification. We applied both forward selection (FS) and backward elimination (BE). Forward selection starts with an empty feature set. New features are added to this set in a greedy manner, when a feature, which can increase accuracy the most, is added. Backward elimination starts with a feature set which contains all considered features. Features that do not decrease accuracy get removed from the feature set, i.e. they get eliminated.

Table V
KDD99 TRAINING DATA SET: RESULTS.

Classifier	Acc [%]	# features	ξ
WAODE+FS	98.99±0.02	2	0.01902
HNB+FS	98.95±0.03	4	0.01911
RBFN1+FS	98.95±0.02	4	0.01945
RBFN2+FS	98.96±0.04	4	0.01914
RBFN3+FS	98.83±0.13	3	0.02094
RBFN5+FS	98.80±0.19	5	0.02135
NB+FS	98.59±0.01	7	0.08713
NB+BE	98.58±0.02	12	0.09235

The classifiers that we applied after nominal feature transformation and feature selection are:

- Weka WAODE (Weightily Averaged One-Dependence Estimators) [15]. This algorithm assigns different weights to one-dependence classifiers, which overcomes the attribute independence assumption of Naive Bayes by averaging over all models in which all attributes depend upon the class and a single other attribute.
- Weka RBFNetwork implements a normalized Gaussian radial basis function network. It uses k-means clustering algorithm to find the centers for the Gaussian radial basis functions. If a class is nominal, it applies the given number of clusters per class. It normalizes all numeric attributes to zero mean and unit variance. We used four different setup parameters: #1{5;0.05} #2{60;0.05} #3{20;0.0001} #5{20;0.05}, where each tuple reflects the number of clusters to generate and the minimum standard deviation for a cluster. Herein we refer to these four alternatives as RBFN1, RBFN2, RBFN3 and RBFN5.
- Weka HNB (Hidden Naive Bayes classification model) [16].
- X-Means clustering [17] with the minimal number of clusters set to 150, the maximal number of runs for k-Means clustering set to 50 and the maximal number of optimization steps per run set to 150. Similar to Xiang et al. [7], the clusters get labelled according to preset priority. Our priority is set as: R2L, U2R, DOS, probe, normal. If a cluster contains a sample belonging to R2L, then all samples in this cluster are classified as R2L attack. Similar applies to the U2R, DOS, probe, normal classes with decreasing priority. A sample is assigned normal label, if and only if a cluster contains only normal samples.
- Weka J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool.

We applied 10-fold cross-validation when estimating classification performance of these classifiers on the *training* data set. We use the default Rapidminer parameters for these classifiers. Their performance is shown in Table V. # features is the number of selected features after either forward selection or backward elimination is applied.

C. Results

Table VI shows the results when the above discussed classifiers are applied to the *test* data set. FP_{norm} is the false positives rate for the normal class, i.e. it is the probability that any attack class gets classified as normal behavior. FN_{norm} is the probability that normal behavior gets classified as an attack. We can observe that when approaches based on cascading classification get excluded, the approach by Pfahringer offers the least classification cost and the approach by Levin offers the highest accuracy.

Table VI shows the results for cascading classification. We can see that E2LSH* $\odot$ XMeans* offers the lowest classification cost of 0.20788. In terms of accuracy, it offers a slightly worse performance than the approach by Levin. The fact that E2LSH* $\odot$ XMeans* results in a low classification cost can be attributed to the decreased number of samples that reach the K_2 classification stage. Note that XMeans clustering takes advantage of class priorities, a technique that cannot be effectively applied to the complete test data set.

Table VII shows the confusion matrix for E2LSH* $\odot$ XMeans*. The result after each classification stage is presented as E2LSH+XMeans (the number of samples predicted by E2LSH* plus the number of samples predicted in the secondary stage by XMeans*). It can be seen that a large number of R2L samples got correctly classified by XMeans. Note that the highest cost penalty is incurred when R2L samples get classified as normal behavior. This helped keep the classification cost low.

VI. Conclusion

The KDD99 network intrusion data set, despite being the focus of several hundred papers, could not be classified in a way that would offer a decisive advantage over a simple 2-nearest-neighbor classification with undecided samples assigned to the class with lowest misclassification penalty. Applying such an approach leads to the classification cost of 0.2178. For comparison, the classification cost obtained by the KDD99 contest winner was 0.2331. It is clear that the contest participants had only limited time to come up with their solution, therefore these two classification costs must be understood in that context.

Given our results, the fact that 1-nearest-neighbor classification landed the 10th place in the contest comes as no surprise. A surprising factor is that the results obtained by applying 1-nearest-neighbor classification received only limited interest. A plausible explanation can be that the relative success of this classification result was perceived as specific for the KDD99 data set with limited general value.

Table VI
KDD99 Test Data Set: Results. * indicates that the shorter training data set was applied.

Classifier	ξ	Acc [%]	FP_{norm} [%]	FN_{norm} [%]
Pfahringer [3]	0.2331	92.71	8.188	0.55
1-Nearest Neighbor [2]	0.2523	92.33	9.118	0.45
Hoque et al. [18]	0.29999	88.26	5.054	30.46
Levin [19]	0.2356	$\boxed{92.92}$	8.475	0.58
XMeans*	0.25709	84.34	$\boxed{0.566}$	43.51
XMeans	0.27761	84.09	2.437	44.25
Weka J48	0.24007	92.60	8.804	0.51
WAODE+FS	0.26480	91.83	9.387	0.64
HNB+FS	0.26395	91.83	9.424	0.66
NB+FS	0.63164	57.12	2.032	84.99
NB+BE	0.27190	91.47	9.374	2.55
RBFN3+FS	0.26365	91.90	9.347	0.68
E2LSH $\odot$ XMeans	0.22711	92.14	6.745	2.61
E2LSH $\odot$ WAODE+FS	0.25799	92.19	9.151	0.26
E2LSH $\odot$ HNB+FS	0.25727	92.18	9.173	0.27
E2LSH $\odot$ NB+FS	0.23479	91.36	6.428	2.84
E2LSH $\odot$ NB+BE	0.26335	91.92	9.104	1.77
E2LSH $\odot$ RBFN3+FS	0.25683	92.25	9.111	0.30
E2LSH* $\odot$ Weka J48	0.24343	92.52	8.950	0.48
E2LSH* $\odot$ XMeans*	$\boxed{0.20788}$	92.27	4.760	2.85
E2LSH* $\odot$ WAODE+FS	0.25697	92.24	9.084	$\boxed{0.23}$
E2LSH* $\odot$ HNB+FS	0.25622	92.24	9.106	0.25
E2LSH* $\odot$ NB+FS	0.23475	91.29	6.303	3.37
E2LSH* $\odot$ NB+BE	0.26241	91.96	9.037	1.77
E2LSH* $\odot$ RBFN3+FS	0.25582	92.31	9.044	0.27

Table VII
Confusion matrix for E2LSH* $\odot$ XMeans*: undecided samples = 5.75%, $\xi = 0.20788$.

		Predicted				
		normal	probe	DOS	U2R	R2L
Actual	normal	57956+920	8+923	22+403	+167	+204
	probe	2+159	2218+645	3+732	+3	+404
	DOS	275+116	+4996	221322+2262	0	+882
	U2R	5+54	+33	+57	0	+79
	R2L	10454+857	+33	+3145	+16	+1684

As discussed above, our primary classification approach was based on 2-nearest-neighbor classification which produced a number of undecided samples. Our goal was to design a secondary classification procedure so that we can obtain a lower classification cost than 0.2178. We applied a range of classification approaches. The lowest classification cost 0.20788 was obtained with X-Means clustering. Note that when X-Means clustering gets applied in isolation (without 2-nearest-neighbor classification), the obtained cost is 0.25709.

We applied cascading classification which only received limited attention from the research community. Interestingly, two classifiers, each unsuitable if applied in isolation, when applied in a cascade lead to a low classification cost. This shows that cascading classification is a viable option, especially, in scenarios where complex machine algorithms such as neural networks or support vector machines cannot be applied, for example due to performance considerations on resource constrained devices such as mobile phones.

Acknowledgment

The authors were supported by the Competence Center of Intelligent Technologies for Electronization and Informatization of Systems and Services (ITMS 26240220072) co-funded by the European Regional Development Fund (ERDF): ”we support research activities in Slovakia / this project is being co-financed by the European Union” – www.asfeu.sk.

References

[1] M. Drozda, I. Bate, and J. Timmis, “Bio-inspired error detection for complex systems,” in *Proceedings of 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2011, pp. 154–163. [Online]. Available: http://dx.doi.org/10.1109/PRDC.2011.27

[2] C. Elkan, “Results of the KDD’99 classifier learning,” *SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 63–64, Jan. 2000. [Online]. Available: http://doi.acm.org/10.1145/846183.846199

[3] B. Pfahringer, “Winning the KDD99 classification cup: Bagged boosting,” *SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 65–66, Jan. 2000. [Online]. Available: http://doi.acm.org/10.1145/846183.846200

[4] S. Marcek, M. Drozda, G. Juhas, and F. Lehocki, “Network intrusion detection in high dimensional space,” in *Proceedings of 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, 2009, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1109/ISABEL.2009.5373652

[5] L. Koc, T. A. Mazzuchi, and S. Sarkani, “A network intrusion detection system based on a hidden naïve bayes multiclass classifier,” *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13 492–13 500, Dec. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.07.009

[6] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, “Filter methods for feature selection - a comparative study,” in *Proceedings of Intelligent Data Engineering and Automated Learning - IDEAL 2007*, ser. Lecture Notes in Computer Science, H. Yin, P. Tino, E. Corchado, W. Byrne, and X. Yao, Eds. Springer Berlin Heidelberg, 2007, vol. 4881, pp. 178–187. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77226-2_19

[7] C. Xiang, P. C. Yong, and L. S. Meng, “Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees,” *Pattern Recognition Letters*, vol. 29, no. 7, pp. 918–924, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2008.01.008

[8] C. Kaynak and E. Alpaydin, “Multistage cascading of multiple classifiers: One man’s noise is another man’s data,” in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 455–462. [Online]. Available: http://dl.acm.org/citation.cfm?id=645529.658130

[9] J. Gama and P. Brazdil, “Cascade generalization,” *Machine Learning*, vol. 41, no. 3, pp. 315–343, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1007652114878

[10] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007, Accessed: Dec. 24, 2013. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[11] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” in *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006, pp. 459–468. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2006.49

[12] ——, *E2LSH 0.1 user manual*, 2005, Accessed: Dec. 24, 2013. [Online]. Available: http://web.mit.edu/andoni/www/LSH/index.html

[13] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, “Yale: Rapid prototyping for complex data mining tasks,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2006, pp. 935–940. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150531

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: An update,” *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[15] L. Jiang and H. Zhang, “Weightily averaged one-dependence estimators,” in *Proceedings of Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, ser. Lecture Notes in Computer Science, Q. Yang and G. Webb, Eds. Springer Berlin Heidelberg, 2006, vol. 4099, pp. 970–974. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36668-3_116

[16] H. Zhang, L. Jiang, and J. Su, “Hidden naive bayes,” in *Proceedings of The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference (AAAI)*, M. M. Veloso and S. Kambhampati, Eds. AAAI Press / The MIT Press, 2005, pp. 919–924.

[17] D. Pelleg and A. W. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734. [Online]. Available: http://dl.acm.org/citation.cfm?id=645529.657808

[18] M. S. Hoque, M. Mukit, M. Bikas, A. Naser *et al.*, “An implementation of intrusion detection system using genetic algorithm,” *arXiv preprint arXiv:1204.1336*, 2012.

[19] I. Levin, “KDD-99 classifier learning contest LLSoft’s results overview,” *SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 67–75, Jan. 2000. [Online]. Available: http://doi.acm.org/10.1145/846183.846201