# An Ensemble Approach to Classifier Construction based on Bootstrap Aggregation

Dewan Md. Farid
Department of CSE
Jahangirnagar University
Dhaka-1342, Bangladesh

Mohammad Zahidur Rahman
Department of CSE
Jahangirnagar University
Dhaka-1342, Bangladesh

Chowdhury Mofizur Rahman
Department of CSE
United International University
Dhaka-1209, Bangladesh

## ABSTRACT

In this paper, we introduce a new approach to the classification of streaming data based on bootstrap aggregation (bagging). The proposed approach creates an ensemble model by using ID3 classifier, naïve Bayesian classifier, and k-Nearest-Neighbor classifier for a learning scheme where each classifier gives the weighted prediction. ID3, naïve Bayesian, and k-Nearest-Neighbor classifiers are very well known data mining methods, which have been already used in many real life classification problems. The proposed approach addresses the practical problems of the classification of streaming data and successfully tested on a number of benchmark problems including large intrusion detection dataset from the UCI machine learning repository to produce a comparison with the established approaches. The experimental results demonstrate that the proposed ensemble classifier achieved high classification rates and generated very low misclassification error.

## Keywords

Bagging, ID3 Classifier, Naïve Bayesian Classifier, k-Nearest-Neighbor Classifier, Classification Rate.

## 1. INTRODUCTION

Bootstrap aggregation (bagging) combines a series of classifiers to improve the data mining process in supervised learning of classification as well as prediction [1]-[3]. Data mining and knowledge discovery from data (KDD) is the process of extracting knowledge from large amounts of data [4], and have been successfully applied to different classification tasks including, but not limited to, decision making, fault detection, pattern recognition, weather forecasting and image processing. Extracting knowledge from data aims at building a model from the data to predict the future behavior. Bagging is a special case of the model averaging approach. As the classifier task is to map the set of attributes of sample data onto a set of class labels, bagging has become one of the alternative frameworks for classifier design together with the more established data mining methods. In the data mining literature, there are several bootstrap methods to address the classification problems, but commonly used one is the 0.632 bootstrap. For a given data set of $d$ examples, each example has a probability $1\text{-}(1\text{-}1/d)^d$ of being selected at least once. If $d$ is large, the probability approaches $1\text{-}1/e = 0.632$, which means 63.2% of the original examples will end up in the bootstrap, and the remaining 36.8% will form the test set (hence, the name, 0.632 bootstrap) [2]. The bootstrap method works as a method of increasing accuracy based on a majority voting. Also it works well in small data sets. Currently, large quantities of information are generated every day by the users of the Internet, and sensors in automated computer systems even for a small network or industry. For this reason, classification of huge amounts of data becomes a very challenging task. Many data mining algorithms: decision tree, naïve Bayesian classifier, k-Nearest-Neighbor, neural network, support vector machines, and genetic algorithms etc have been applied for classification of huge amounts of data in the last decades [5]-[9]. Depending on the kinds of data to be mined, the data mining system can be categorized according to the applications they adapt. For example, data mining systems can be tailored specifically for finance, stock, telecommunications, DNA, and so on. It has been successfully tested that the bagged classifier of data mining always has improved accuracy over a single classifier.

In this paper, we present a new ensemble classifier based on bootstrap aggregation (bagging), which combines ID3, naïve Bayesian (NB), and k-Nearest-Neighbor (kNN) classifiers to increase the classification rates. The proposed approach first creates a training dataset from a given dataset using selection with replacement technique. It is very likely that some of the examples from the dataset will occur more than once in the training dataset. The examples that did not make it into the training dataset end up forming the test dataset. Then each classifier (ID3, NB, and kNN) creates a classification model from the training examples, and initialized the weight of each classifier based on the accuracies of percentage of correctly classified examples from training dataset. To classify the testing examples or unknown examples each classifier returns its class prediction, which counts as one vote. The proposed bagged classifier counts the votes with the weights of classifiers, and assigns the class with the maximum weighted vote. We tested the performance of the proposed algorithm on a number of benchmark problems including large intrusion detection dataset from the UCI machine learning repository to produce a comparison with other established data mining methods, and the experimental results proved that the proposed algorithm increases the classification rates and reduces the misclassification errors.

The remainder of the paper is organized as follows. Section 2 describes the data mining algorithms: ID3, naïve Bayesian (NB) classifier, and k-Nearest-Neighbor (kNN) classifier. Section 3 presents the proposed algorithm for classifier construction. Section 4 describes the experimental results based on a number of widely used benchmark datasets, as well as using intrusion detection data from the UCI machine learning repository [10]. Finally, Section 5 presents our conclusions with future works.

## 2. MINING ALGORITHMS

The goal of supervised learning in data mining is to identify a function from some input attributes to some output attributes on the basis of observations of the values of the attributes from the training data. The task of supervised learner or classifier is to predict the value of the function for any valid input object after having seen only a small number of training examples. Supervised learning are used in the field of bioinformatics, handwriting recognition, information retrieval, object recognition in computer vision, optical character recognition, pattern recognition, and speech recognition etc.

### 2.1 ID3 Classifier

ID3 uses information gain as its attributes selection measure [11]. The attribute with the highest information gain is chosen as the splitting attribute for node *N*. This attribute minimizes the information needed to classify the examples in the resulting partitions and reflects the least randomness or "impurity" in these partitions. The expected information needed to classify an example in dataset *D* is given by

$$Info(D) = -\sum_{i}^{m} p_i \log_2(p_i) \qquad (1)$$

Where $p_i$ is the probability that an arbitrary example in dataset *D* belongs to class $C_i$ and is estimated by $|C_{i,D}|/|D|$. A log function to the base 2 is used, because the information is encoded in bits. *Info(D)* is just the average amount of information needed to identify the class label of an example in dataset *D*. Partitioning (e.g., where a partition may contain a collection of examples from different classes rather than from a single class) to produce an exact classification of the examples by

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j) \qquad (2)$$

The term $|D_j|/|D|$ acts as the weight of the *j*th partition. $Info_A(D)$ is the expected information required to classify an example from dataset *D* based on the partitioning by *A*. The information gain is defined as the difference between the original information requirement and the new requirement, that is,

$$Gain(A) = Info\ (D) - Info_A(D) \qquad (3)$$

The attribute *A* with the highest information gain, *Gain*(*A*) is chosen as the splitting attribute at node *N*.

### 2.2 Naïve Bayesian Classifier

Naïve Bayesian (NB) classifier is a simple probabilistic classifier [12]-[14]. NB classifier is based on probability models that incorporate strong independence assumptions which often have no bearing in reality, hence are (deliberately) naïve. A more descriptive term for the underlying probability model would be independent feature model. Furthermore the probability model can be derived using Bayes' Theorem. NB classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label *c*. The conditional independence assumption can be formally stated as follows:

$$P(A \mid C = c) = \prod_{i=1}^{n} P(A_i \mid C = c) \qquad (4)$$

Where each attribute set $A = \{A_1, A_2, \ldots., A_n\}$consists of *n* attribute values. With the conditional independence assumption, instead of computing the class-conditional probability for every combination of *A*, only estimate the conditional probability of each $A_i$, given *C*. The latter approach is more practical because it does not require a very large training set to obtain a good estimate of the probability. To classify a test example, the naïve Bayesian classifier computes the posterior probability for each class *C*.

$$P(C \mid A) = \frac{P(C)\prod_{i=1}^{n} P(A_i \mid C)}{P(A)} \qquad (5)$$

Since *P(A)* is fixed for every *A*, it is sufficient to choose the class that maximizes the numerator term,

$$P(C)\prod_{i=1}^{n} P(A_i \mid C) \qquad (6)$$

The naïve Bayesian classifier has several advantages. It is easy to use, and unlike other classification approaches, only one scan of the training data is required. The naïve Bayesian classifier can easily handle missing attribute values by simply omitting the probability when calculating the likelihoods of membership in each class. The NB classifier is straightforward to use, where there are simple relationships, it often does yield good results.

### 2.3 K-Nearest-Neighbor Classifier

The k-Nearest-Neighbor (kNN) uses the distance measure techniques [15], [16]. kNN finds *k* examples in training data that are closest to the test example and assigns the most frequent class label among the training examples to the test example. When a classification is to be made for a new example, its distance to each attribute in the training data must be determined. Only the *k* closest examples in the training data are considered further. "Closest" is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or examples, say, $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$, is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \qquad (7)$$

For, k-nearest-neighbor classification, the unknown example is assigned the most common class among its k nearest neighbors.

## 3. PROPOSED BAGGING ALGORITHM

Given a dataset $D$, of $d$ examples and the dataset $D$ contains the following attributes $\{A_1, A_2,...,A_n\}$ and each attribute $A_i$ contains the following attribute values $\{A_{i1}, A_{i2},...,A_{ih}\}$. Also the dataset $D$ contains a set of classes $C = \{C_1, C_2,...,C_m\}$, where each example in dataset $D$ has a particular class $C_j$. The algorithm first generates the training dataset $D_i$ from the given dataset $D$ using selection with replacement technique. It is very likely that some of the examples from the dataset $D$ will occur more than once in the training dataset $D_i$. The examples that did not make it into the training dataset end up forming the test dataset. Then a classifier model, $M_i$, is learned for each training examples $d$ from training dataset $D_i$. The algorithm builds three classifiers using ID3, naïve Bayesian (NB), and k-Nearest-Neighbor (kNN) classifiers.

The basic strategy used by ID3 classifier is to choose splitting attributes with the highest information gain first and then builds a decision tree. The amount of information associated with an attribute value is related to the probability of occurrence. The concept used to quantify information is called entropy, which is used to measure the amount of randomness from a data set. When all data in a set belong to a single class, there is no uncertainty, and then the entropy is zero. The objective of decision tree classification is to iteratively partition the given data set into subsets where all elements in each final subset belong to the same class. The entropy calculation is shown in equation 8. Given probabilities $p_1, p_2,..,p_s$ for different classes in the data set

$$Entropy: H(p_1,p_2,...p_s) = \sum_{i=1}^{s} (p_i \log(1/p_i)) \tag{8}$$

Given a data set, $D$, $H(D)$ finds the amount of entropy in class based subsets of the data set. When that subset is split into s new subsets $S = \{D_1, D_2,...,D_s\}$ using some attribute, we can again look at the entropy of those subsets. A subset of data set is completely ordered and does not need any further split if all examples in it belong to the same class. The ID3 algorithm calculates the information gain of a split by using equation 9 and chooses that split which provides maximum information gain.

$$Gain\ (D,S) = H(D)-\sum_{i=1}^{s} p(D_i)H(D_i) \tag{9}$$

The naïve Bayesian (NB) classifier calculates the prior probability, $P(C_j)$ and class conditional probability, $P(A_{ij}|C_j)$ from the dataset. For classifying an example, the NB classifier uses these prior and conditional probabilities to make the prediction of class for that example. The prior probability $P(C_j)$ for each class is estimated by counting how often each class occurs in the dataset $D_i$. For each attribute $A_i$ the number of occurrences of each attribute value $A_{ij}$ can be counted to determine $P(A_i)$. Similarly, the class conditional probability $P(A_{ij}|C_j)$ for each attribute values $A_{ij}$ can be estimated by counting how often each attribute value occurs in the class in the dataset $D_i$.

The k-Nearest-Neighbor (kNN) classifier assumes that the entire training set includes not only the data in the set but also the desired classification for each item. When a classification is to be made for a test or new example, its distance to each item in the training data must be determined. The test or new example is then placed in the class that contains the most examples from this training data of k closest items.

After building classifiers using ID3, NB, and kNN, each classifier, $M_i$, classifies the training examples and initialized the weight, $W_i$ of each classifier based on the accuracies of percentage of correctly classified examples from training dataset. To classify the testing examples or unknown examples each classifier returns its class prediction, which counts as one vote. The proposed bagged classifier counts the votes with the weights of classifiers, and assigns the class with the maximum weighted vote. The main procedure of proposed algorithm is described as follows:

**Algorithm:** An ensemble classifier using bagging, ID3, naïve Bayesian classifier, and k-Nearest-Neighbor.

Input:

- $D$, a set of $d$ examples.
- $k = 3$, the number of models in the ensemble.
- Learning scheme (ID3, naïve Bayesian classifier, and k-Nearest-Neighbor)

Output: A composite model, $M^*$.

Procedure:

```
1. Generate a new training dataset $D_i$ with equal number of
   examples from a given dataset $D$ using selection with
   replacement technique. Same example from given dataset $D$
   may occur more than once in the training dataset $D_i$.
2. **for** $i = 1$ to $k$ **do**
3. Derive a model or classifier, $M_i$ using training dataset $D_i$.
4. Classify each example $d$ in training data $D_i$ and initialized
   the weight, $W_i$ for the model, $M_i$, based on the accuracies of
   percentage of correctly classified example in training data
   $D_i$.
5. **endfor**
```

To use the composite model on test examples or unseen examples:

```
1. **for** $i = 1$ to $k$ **do**
2. Classify the test or unseen examples using the $k$ models.
3. Returns a weighted vote (which counts as one vote).
4. **endfor**
5. $M^*$, counts the votes and assigns the class with the
   maximum weighted vote for that example.
```

## 4. EXPERIMENTAL ANALYSIS

The proposed bagging classifier was tested on a number of widely used benchmark problems from the UCI machine learning repository [10] and on a large data stream of intrusion detection data [17]. The main reason of using datasets from UCI repository is that a number of solutions exist in the literature for classification.

### 4.1 Benchmark Problems from UCI Repository

1. Tic-Tac-Toe: It encodes the complete set of possible board configurations at the end of Tic-Tac-Toe games, where “x” is assumed to play first. The target concept is “win of x” (i.e., true when “x” has one of 8 possible ways to create a “three-in-a-row”). In Tic-Tac-Toe dataset, there are total 958 examples (626 positive examples and 332 negative examples), number of classes: 2 (positive and negative), and the number of attributes: 9 (each attribute corresponding to one tic-tac-toe square and has 3 attribute values x, o, and b)

2. Soybean: There are 19 classes, only the first 15 of which have been used in prior work. The folklore seems to be that the last four classes are unjustified by the data since they have so few examples. There are 35 categorical attributes and total 683 examples in this dataset.

3. Iris: This data set contains 3 classes and total 151 examples, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

4. Zoo: A simple database containing 16 Boolean-valued attributes with 101 examples. The "type" attribute appears to be the class attribute. Here is a breakdown of which animals are in which type: (I find it unusual that there are 2 instances of "frog" and one of "girl"!).

5. Diabetes: This data set prepared for the use of participants for the 1994 AAAI Spring Symposium on Artificial Intelligence in Medicine. Diabetes patient records were obtained from two sources: an automatic electronic recording device and paper records. The automatic device had an internal clock to timestamp events, whereas the paper records only provided "logical time" slots (breakfast, lunch, dinner, bedtime). There are 2 classes, 8 attributes and total 768 examples in this dataset.

The results are listed in table 1.

**Table 1. Classification Rates (%) in Different Domains**

| Datasets | ID3 | NB | kNN | Proposed Classifier |
|---|---|---|---|---|
| Tic-Tac-Toe | 86.39 | 98.43 | 94.32 | 100 |
| Soybean | 91.23 | 97.65 | 89.95 | 100 |
| Iris | 95.45 | 92.43 | 82.46 | 99.85 |
| Zoo | 92.67 | 98.69 | 90.12 | 99.59 |
| Diabetes | 73.17 | 96.70 | 77.19 | 100 |

### 4.2 Intrusion Detection Dataset Stream

The KDD cup 1999 dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector, a predictive model capable of distinguishing between intrusions and normal connections. There are total 41 attributes in KDD99 dataset for each network connection that have either discrete or continuous values and divided into three groups. The first group of attributes is the basic features of network connection, which include the duration, prototype, service, number of bytes from source IP addresses or from destination IP addresses, and some flags in TCP connections. The second group of attributes in KDD99 is composed of the content features of network connections and the third group is composed of the statistical features that are computed either by a time window or a window of certain kind of connections. The classes in KDD99 dataset can be categorized into five main classes (one normal class and four main intrusion classes: probe, DOS, U2R, and R2L). Table 2 shows the comparison of the results for the KDD99 intrusion detection dataset.

**Table 2. Comparison of the Results for the Intrusion Detection Dataset (detection rate %)**

| Method | Normal | Probe | DOS | U2R | R2L |
|---|---|---|---|---|---|
| ID3 | 99.63 | 97.85 | 99.51 | 49.21 | 92.75 |
| NB | 99.27 | 99.11 | 99.69 | 64.00 | 99.11 |
| kNN | 99.60 | 75.00 | 97.30 | 35.00 | 0.60 |
| Proposed Classifier | 100 | 99.92 | 99.93 | 99.57 | 99.61 |

## 5. CONCLUSIONS & FUTURE WORKS

This paper introduced a new bootstrap aggregation (bagging) based classifier that ensembles ID3 classifier, naïve Bayesian classifier, and k-Nearest-Neighbor classifier. The proposed classifier can improve the classification rates and reduces the misclassification error rates. It is already successfully tested that the bagging classifier always improves the classification rate over a single classifier. The proposed classifier used ID3, naïve Bayesian, and k-Nearest-Neighbor classifiers, which are very popular and useful data mining algorithms for supervised learning. We tested the performance of proposed classifier on six benchmark datasets from UCI machine learning repository including intrusion detection problem, and the results prove that the proposed new classifier achieves high classification rates for different datasets. The future research issue will be applying this classifier in classification problems of real world problem domains.

## 6. ACKNOWLEDGMENTS

Support for this research received from Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh, and Department of Computer Science and Engineering, United International University, Bangladesh.

## 7. REFERENCES

[1] L. Breiman, “Bagging predictors,” Machine Learning, Vol. 24, 1996, pp. 123-140.

[2] B. Efron, and R. Tibshirani, “An Introduction to the Bootstrap,” Chapman & Hall, 1993.

[3] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” In Proc. of the 14th Joint International Conference Artificial Intelligence (IJCAI’95), Vol. 2, August 1995, Montreal, Canada, pp. 1137-1143.

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview," In Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.

[5] S. Pang, S. Ozawa, and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams," *IEEE Trans. Syst. Man Cybern. B*, Cybern., Vol. 35, No. 5, Oct. 2005, pp. 905-914.

[6] R. Jin, and G. Agrawal, "Efficient decision tree construction on streaming data," in Proc. *ACM SIGKDD*, 2003, pp. 571-576.

[7] L. Breiman, J. Friedman, C. Stone, and R. Olshen, "Classification and Regression Trees," Boca Raton, Fl: Chapman & Hall, 1993.

[8] K. M. A. Chai, H. T. Ng, and H. L. Chieu, "Bayesian online classifiers for text classification and filtering," in Proc. *SIGIR* 2002, Tampere, Finland, Aug. 11-15, pp. 97-104.

[9] C. M. Bishop, "Neural Networks for Pattern Recognition," Oxford, U.K.: Oxford University Press, 1995.

[10] The Archive UCI Machine Learning Datasets. http://archive.ics.uci.edu/ml/datasets/

[11] J. R. Quinlan, "Induction of Decision Tree," Machine Learning Vol. 1, pp. 81-106, 1986.

[12] P. Langley, "Induction of recursive Bayesian classifier," *In Proc. of the European Conference on Machine Learning*, 1993, pp. 153-164.

[13] P. Langely, W. Iba, and K. Thomas, "An analysis of Bayesian classifier," *In Proc. of the 10th National Conference on Artificial Intelligence*, San Mateo, CA: AAAI Press, 1992, pp. 223-228.

[14] Z. Zheng, and I. G. Webb, "Lazy learning of Bayesian roles," *Machine Learning-1*, Kluwer Academic Publishers, Boston, 2000, pp. 1-35.

[15] B. V. Dasarathy, "Nearest Neighor (NN) Norms: NN Pattern Classification Techniques," IEEE Computer Society Press, 1991,

[16] Duda, R., P.E. Hart, and D.G. Stork, "Pattern classification," Second edn. John Wiley & Sons, 2001.

[17] The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

## 8. AUTHORS PROFILE

**Dewan Md. Farid** is a doctoral candidate in the Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh. He obtained B.Sc. Engineering in Computer Science and Engineering from Asian University of Bangladesh in 2003 and Master of Science in Computer Science and Engineering from United International University, Bangladesh in 2004. He is a part-time faculty member in the Department of Computer Science and Engineering, United International University, Bangladesh. He is a member of IEEE and IEEE Computer Society. He has published 5 international journals and 9 international conference papers in the field of data mining, machine learning, and intrusion detection. He has participated and presented his papers in international conferences at France, Italy, Portugal, and Malaysia. He worked as a visiting researcher at ERIC Laboratory, University Lumière Lyon 2 – France from 01-09-2009 to 30-06-2010.

**Mohammad Zahidur Rahma** is currently a Professor at Department of Computer Science and Engineering, Jahangirnager University, Banglasesh. He obtained his B.Sc. Engineering in Electrical and Electronics from Bangladesh University of Engineering and Technology in 1986 and his M.Sc. Engineering in Computer Science and Engineering from the same institute in 1989. He obtained his Ph.D. degree in Computer Science and Information Technology from University of Malaya in 2001. He is a co-author of a book on E-commerce published from Malaysia. His current research includes the development of a secure distributed computing environment and e-commerce.

**Professor Dr. Chowdhury Mofizur Rahman** had his B.Sc. (EEE) and M.Sc. (CSE) from Bangladesh University of Engineering and Technology (BUET) in 1989 and 1992 respectively. He earned his Ph.D from Tokyo Institute of Technology in 1996 under the auspices of Japanese Government scholarship. Prof Chowdhury is presently working as the Pro Vice Chancellor and acting treasurer of United International University (UIU), Dhaka, Bangladesh. He is also one of the founder trustees of UIU. Before joining UIU he worked as the head of Computer Science & Engineering department of Bangladesh University of Engineering & Technology which is the number one technical public university in Bangladesh. His research area covers Data Mining, Machine Learning, AI and Pattern Recognition. He is active in research activities and published around 100 technical papers in international journals and conferences. He was the Editor of IEB journal and worked as the moderator of NCC accredited centers in Bangladesh. He worked as the organizing chair and program committee member of a number of international conferences held in Bangladesh and abroad. At present he is acting as the coordinator from Bangladesh for EU sponsored eLINK project. Prof Chowdhury has been working as the external expert member for Computer Science departments of a number of renowned public and private universities in Bangladesh. He is actively contributing towards the national goal of converting the country towards Digital Bangladesh.