



Ensemble of Decision Tree Classifiers for Mining Web Data Streams

Fauzia Yasmeen Tani
Department of CSE
Jahangirnagar University
Dhaka-1342, Bangladesh

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

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

ABSTRACT

The World Wide Web (www or w3 commonly known as the web) is the largest database available with growth at the rate of millions of pages a day and presents a challenging task for mining web data streams. Currently extraction of knowledge from web data streams is getting more and more complex, because the structure of data doesn't match the attribute-values when considering the large volume of web data. In this paper, an ensemble of decision tree classifiers is presented, which is an efficient mining method to obtain a proper set of rules for extracting knowledge from a large amount of web data streams. We built a web server using Model 2 Architecture to collect the web data streams and applied the ensemble classifier for generating decision rules using several decision tree learning models. Experimental results demonstrate that the proposed method performs well in decision making and predicting the class value of new web data streams.

Keywords

Data Streams, Decision Tree, J2EE, Model 2 Architecture, Web Server, and Web Mining.

1. INTRODUCTION

Data mining and knowledge discovery in databases (KDD) is the process of finding hidden information in a database, and web mining is the mining of web related data to the World Wide Web. Or web mining is the application of data mining algorithms to discover hidden patterns from the web. Web mining can be divided into three different types, which are web usage mining, web content mining and web structure mining. Web usage mining is the process of extracting useful information from web data streams. Web data streams are the ordered sequence of instances generated by the web server. Web content mining is the process of extraction and integration of useful data, information and knowledge from web page contents, which performed by basic search engine. Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site/application.

The decision tree (DT) induction is most efficient and well known mining method in classification problem of supervised learning [1]-[4], which has been already applied in many real world decision making applications like medical diagnosis, radar signal classification, credit approval, weather prediction, fraud detection, and customer segmentation etc. There are many advantages to the use of decision trees for classification task. Classification referred to as supervised learning because it maps

the data into predefined classes before examining the data. Rules generation using decision tree are easy to understand, the tree size is independent of the database size, and the tree can be constructed for large dataset with many attributes [5]. In early 1980s, J. Ross Quinlan introduced the decision tree learning algorithm, which is known as ID3 (Iterative Dichotomiser) algorithm [6], [7]. In 1984, L. Breiman et al. presented binary decision tree by publishing the book Classification and Regression Tree (CART), which applied to knowledge acquisition for expert systems where an expert supplies examples and the resulting decision tree may be used in the formulation of rules [8]. Later in 1993, J. Ross Quinlan again introduced C4.5 decision tree learning algorithm, which is the successor of ID3 algorithm [9]. C4.5 supervised learning algorithm become very popular among other machine learning algorithms in that time. Traditional decision tree building algorithms such as ID3, C4.5, and CART rely on a greedy (i.e., none backtracking) approach in which trees are constructed in a top-down recursive divide-and-conquer manner. A decision tree is built from a training dataset by recursively partitioning training dataset [10], [11]. A training dataset consists of set of attributes and classes. Attributes values can be continuous or discrete. Once the tree is built, it's applied to classify unknown examples, whose attribute values are known, but class value is unknown.

The decision tree algorithms use a divide and conquer approach to construct a tree. During the tree construction, attribute selection measures are used to select the best attribute, which can partition the examples of dataset into distinct classes [12], [13]. In the past decade, many attribute selection heuristics have been introduced by the researchers such as information gain, gain ratio, and gini index [14]. The choice of attribute involves not only an examination of the data in the training set but also the informed input of domain experts. In this paper, an ensemble of decision tree classifiers is introduced for mining web data streams. We built a J2EE web application to collect the web data streams. Then generated three training datasets with equal number of examples using web data streams, and built three decision tree models using each training dataset. For classifying the test or unseen examples: counts the weighted votes for each decision tree and assigns the class with the maximum weighted vote for that example.

The rest of the paper is organized as follows. Section 2 describes the Model 2 Architecture. Section 3 presents the decision tree methods. Section 4 describes the experimental setup and analysis. Finally, Section 5 draws conclusions.

2. MODEL 2 ARCHITECTURE

The Model 2 architecture for designing the web applications has three features: Model, View, and Controller that is known as MVC diagram. Model implements the business logics of the web application, which are the Java technology classes. Business logic is anything that manipulates data to accomplish something, such as storing data. Presentation logic is anything that controls how the application presents information to the user. Generating the HTML response within the servlet code is presentation logic. View is the HTML (Hypertext Markup Language) response, retrieving data from the model necessary to generate the response, and provides HTML forms to permit user interaction. HTML is a document display language that lets users link from one document to another. HTML also permits images and other media objects to be embedded in an HTML document. Controller verifies form data, updates the model with the form data, and selects the next view as the response. The Java servlet acts as the controller in model 2 architecture. The Java servlet runs as a thread in the web container. When the number of requests for a servlet rises, no additional instances of the servlet or operating system processes are created. Each request is processed concurrently using one Java thread per request. A Java servlet is a Java technology program that can recognize HTTP (Hypertext Transfer Protocol) requests, generate the response dynamically, and then send a response containing a HTML page or document to the browser. HTTP is a protocol to transfer files from the web server to the web browser. It supports only one request per connection, which means that with HTTP the client/browser connects to the web server to retrieve one file and then disconnects. This mechanism allows more users to connect to a given web server over a period to time.

A servlet is a Java class in Java EE that conforms to the Java Servlet API, a protocol by which a Java class may respond to requests. Java servlets run within component container architecture. This container is called the web container. The web container is a Java Virtual Machine (JVM) that supplies an implementation of the servlet API. JVM is an imaginary machine that is implemented by emulating it in software on a

real machine. Code for the JVM is stored in .class files, each of which contains code for at most one public class. Following shows the advantages of Java servlets:

- Each request is run in a separate thread.
- Servlets are scalable.
- Servlets are robust and Object oriented.
- Servlets can only be written in the Java programming language.
- Servlets are platform independent.
- Servlets have access to logging capabilities.
- The web container provides additional services to the servlets, such as error handling and security.

Servlet and JSP (JavaServer Pages) pages are an integral part of the J2EE platform. JSP pages are just one way of implementing the concept of HTML pages. With JSP pages, Java technology code fragments are embedded in an HTML-like file. JSP pages are converted by the web container into a servlet instance. That servlet then processes each request to that JSP page. The JSP page runs as a servlet. J2EE platform is container based on Enterprise JavaBeans (EJB) technology. An EJB container provides a modular framework for business logic components based on EJB architecture called enterprise beans. An EJB container also provides services for transaction management, persistence, security, and life cycle management of enterprise beans.

Model-View-Controller (MVC) is the Blueprint for J2EE web application design that is easier to maintain for big web application project. MVC clearly defines the responsibilities of participating classes, making bugs easier to track down and eliminate. MVC separates design concerns (data persistence and behavior, presentation, and control), decreasing code duplication, centralizing control, and making the application more easily modifiable. The MVC architecture is to separate the business logic and application data from the presentation data to the user. Figure 1 shows Model 2 Architecture.

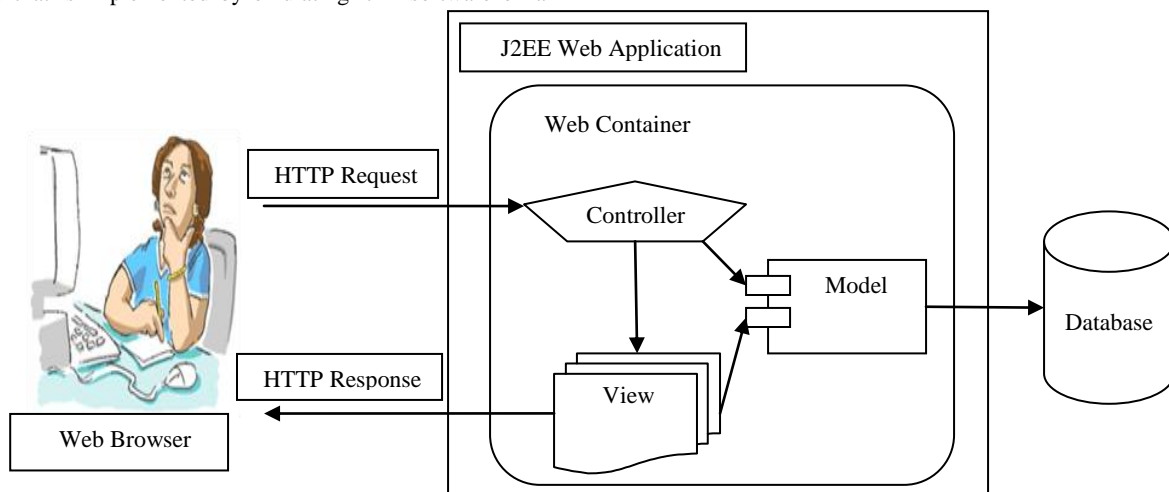


Fig 1: J2EE Web Application Design using Model 2 (MVC) Architecture.

3. DECISION TREE LEARNING

Decision tree learning is a method commonly used in data mining. Decision tree induction is the learning of flowchart-like tree structure from class-labeled training examples. A decision tree has three main components: nodes, leaves, and edges. Each node is labeled with an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to possible values of the attribute. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. To make a decision using a decision Tree, start at the root node and follow the tree down the branches until a leaf node representing the class is reached. Each decision tree represents a rule set, which categorizes data according to the attributes of dataset. The decision tree building algorithms may initially build the tree and then prune it for more effective classification. With pruning technique, portions of the tree may be removed or combined to reduce the overall size of the tree. The time and space complexity of constructing a decision tree depends on the size of the data set, the number of attributes in the data set, and the shape of the resulting tree. The construction of decision tree classifiers does not require any domain knowledge, and can handle high dimensional data. The learning and classification process of decision tree induction is simple and fast.

3.1 Decision Tree Methods

ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree from a fixed set of training examples [6], which is the precursor to the C4.5 algorithm. ID3 searches through the attributes of training instances and extracts the attribute that best separates the given examples. It chooses the best attribute with maximum information gain value that can separate the training dataset [7]. The amount of information associated with an attribute value is related to the probability of occurrence. The resulting tree is used to classify future instances, whose attributes value are known but class value is unknown.

The C4.5 algorithm generates a classification tree by recursive partitioning of training data [9]. The decision is grown using Depth-first strategy. C4.5 tests all the input attributes that can split the training dataset and selects a test input attributes that gives the best information gain. 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 entropy calculation is shown in equation 1. Given probabilities p_1, p_2, \dots, p_s for different classes in the data set

$$\text{Entropy: } H(p_1, p_2, \dots, p_s) = \sum_{i=1}^s (p_i \log(1/p_i)) \quad (1)$$

Given a data set, D , $H(D)$ finds the amount of entropy in class based subsets $S = \{D_1, D_2, \dots, D_s\}$. The information gain is calculated using equation 2.

$$\text{Gain } (D, S) = H(D) - \sum_{i=1}^s p(D_i)H(D_i) \quad (2)$$

The C4.5 algorithm uses highest *Gain Ratio* in equation 3 for splitting purpose that ensures a larger than average information gain.

$$\text{GainRatio}(D, S) = \frac{\text{Gain}(D, S)}{H\left(\frac{|D_1|}{|D|}, \dots, \frac{|D_s|}{|D|}\right)} \quad (3)$$

J48 is slightly modified C4.5 algorithm in Weka. Weka is a collection of machine learning algorithms for solving real-world data mining problems, which contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Weka is written in Java technology and runs on almost any platform. The algorithms can either be applied directly to a dataset or called from our own Java code.

CART (Classification and Regression Trees) is a robust and easy-to-use decision tree that automatically sifts large, complex databases, searching for and isolating significant patterns and relationships [8]. CART handles missing data and contains a pruning strategy. CART was the primary tool used to win the KDDCup 2000 web-mining competition and is currently in use in major web applications. CART's binary decision trees are more sparing with data and detect more structure before too little data are left for learning. Other decision tree approaches use multi-way splits that fragment the data rapidly, making it difficult to detect rules that require broad ranges of data to discover. The SPRINT (Scalable Parallelizable Induction of Decision Trees) algorithm uses an impurity function called *gini* index to find the best split.

$$\text{gini } (D) = 1 - \sum p_j^2 \quad (4)$$

Where, p_j is the probability of class C_j in data set D . The goodness of a split of D into subsets D_1 and D_2 is defined by

$$\text{gini}_{\text{split}}(D) = n_1/n(\text{gini}(D_1)) + n_2/n(\text{gini}(D_2)) \quad (5)$$

A random tree is a tree drawn at random from a set of possible trees. Random trees can be generated efficiently and the combination of large sets of random trees generally leads to accurate models. Random forest is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees. It choosing random subsets of variables for each tree and using the most frequent tree output as the overall classification.

Naïve Bayesian tree (NBTree) is a hybrid learning approach of decision tree and naïve Bayesian (NB) classifier [15]. In NBTree nodes contain and split as regular decision-trees, but the leaves are replaced by NB classifier, the advantage of both decision tree and naïve Bayes can be utilized simultaneously. Depending on the precise nature of the probability model, NB classifier can be trained very efficiently in a supervised learning. NBTree splits the dataset by applying entropy based algorithm and then used standard NB classifiers at the leaf node to handle attributes. It applies strategy to construct decision tree and replaces leaf node with NB classifier.

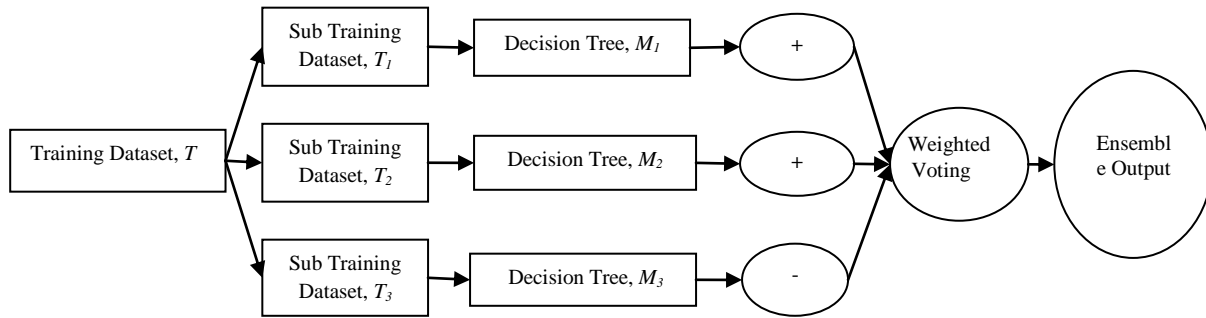


Fig 2: Ensemble of Decision Tree Classifiers.

3.2 Proposed Ensemble Classifier

Given a training dataset T , this contains the data instances. Data instances contain the following attributes $\{A_1, A_2, \dots, A_n\}$ and each attribute A_i contains the following attribute values $\{A_{i1}, A_{i2}, \dots, A_{in}\}$. Also the training dataset contains a set of classes $C = \{C_1, C_2, \dots, C_m\}$, where each instance in training data has a particular class C_j . The algorithm first generates three sub training datasets T_1, T_2 , and T_3 of equal instances from the training dataset T . It is very likely that same instance from the dataset T will occur more in the sub training datasets. Then three decision tree classifiers M_1, M_2 , and M_3 , are learned from their each sub training dataset T_i . The algorithm builds decision tree classifiers using ID3 classifier.

After building the decision tree classifiers using ID3 classifier the algorithm initialized the weight, W_i of each decision tree classifier M_i , based on the accuracies of percentage of correctly classified examples from their respective sub training dataset. To classify the testing examples or new/unknown examples each decision tree classifier returns its class prediction, which counts as one vote. The proposed algorithm counts the votes with the weights of classifiers, and assigns the class with the maximum weighted vote to the new/unseen examples. The figure 2 shows the ensemble of decision tree classifiers. The main procedure of proposed algorithm is described as follows:

Algorithm: Ensemble of Decision Tree Classifiers.

Input:

- T , Training Dataset.
- $k = 3$, the number of models in the ensemble.
- Learning scheme: ID3 classifier.

Output: A composite model, M^* .

Procedure:

1. Generate three new sub training dataset T_1, T_2 , and T_3 with equal number of examples from given training dataset T . Same example from given training dataset T may occur in the sub training datasets T_i .
2. **for** $i = 1$ to k **do**
3. Derive a decision tree model M_i using the sub training dataset T_i .
4. Classify each example of sub training dataset T_i with decision tree model M_i and initialized the weight, W_i for the model, M_i , based on the accuracies of percentage of correctly classified example in sub training dataset T_i .
5. **endfor**
6. For classifying the test or unseen examples: counts the votes for each classifier and assigns the class with the maximum weighted vote for that example.

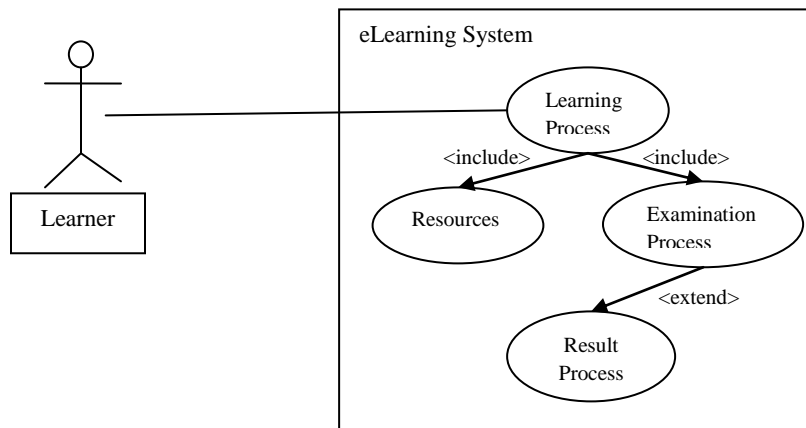


Fig 3: Use Case Diagram for eLearning System.

4. EXPERIMENTAL ANALYSIS

To collect web data streams, we developed an eLearning based web server using J2EE, which implements the Model 2 (MVC) Architecture. J2EE architecture is very popular tools for developing B2B and B2C web applications. We have developed 2 controller servlets, 2 model classes, and 11 view classes. All these model, view, and controller Servlets are written by Java programming language. Figure 3 shows the use case diagram for eLearning system. A Use Case diagram is a simple diagram that uses to document the Requirements Gathering and Analysis work flows. A Use Case diagram represents the functionality provided by the system to external users. The Use Case diagram is composed of actors, use case nodes, and their relationships. Actors can be humans or other systems.

A Class diagrams shows set of classes in the system, and the associations and inheritance relationships between the classes. Class nodes might also contain a listing of attributes and operations. Table 1 and 2 shows the class diagram of Examination and Book class in eLearning system.

Table 1. Class diagram of Examination class in eLearning system.

Examination
- examinerName: String = null
- examID: int = 0
+ result: int[] = {4, 3, 3, 5, 1, 2}
+ cResult: int[]
+ getExaminerName():String
+ getExamID(): int
+ setExaminerName(examinerName: String): void
+ setExamID(examID: int): void

Table 2. Class diagram of Book class in eLearning system.

Book
- bookName : String = null
- authorsName : String = null
- bookID : int = 0
<<constructor>>
+ Book (bookName : String, authorsName : String, bookID : int)
<<accessor>>
+ getBookName() : String
+ getAuthorsName() : String
+ getBookID() : int
< <mutator>>
+ setBookName(bookName : String)
+ setAuthorsName(authorsName : String)
+ setBookID(bookID : int)
<<business logic>>
+ showBook(bookID : int)

Table 3 shows the web data streams, which is the login process of a user that contains input attributes like: AccessTime, NetworkIP, Protocol, Pattern, LogData, and Id, with class attribute: Valid. Class value is either TRUE or FALSE that means a user can successfully login to the system or not. Training dataset is generated by using web stream data.

We generated three training datasets with 300 examples in each dataset and randomly select testing examples of 250 examples. Some of the test examples are not presented in the training dataset. We built three decision tree models using each training dataset and classify the examples of each training examples using their respective decision tree model. Also we calculate the classification accuracy of each decision tree model and initialized the weight on each decision tree model based on their classification rate. Proposed ensemble of decision tree classifier can classify the test examples with 98.4% accuracy.

Table 3. Sample of web data streams.

Access Time	Network IP	Protocol	Pattern	Log Data	Id	Valid
12/19/2011 19:22	192.168.0.11	Http:POST	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	133	FALSE
12/16/2011 22:06	127.0.0.2	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 2	TRUE
12/16/2011 22:27	127.0.0.1	Http:POST	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 3	FALSE
12/16/2011 22:27	127.0.0.2	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 4	TRUE
12/16/2011 22:35	127.0.0.5	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 7	TRUE
12/16/2011 22:28	127.0.0.2	Http:GET	NULL	Opera/9.64 (Windows NT 5.1; U; en) Presto/2.1.1	9 5	TRUE
12/16/2011 22:00	127.0.0.1	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	8 9	TRUE
12/16/2011 22:04	127.0.0.1	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 0	FALSE
12/16/2011 22:06	127.0.0.1	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 1	FALSE
12/16/2011 22:30	192.168.0.1	Http:GET	NULL	Mozilla/5.0 (Windows NT 5.1; rv:2.0b12) Gecko/20100101 Firefox/4.0b12	9 6	TRUE



Table 4. Classification rate (%) of decision tree models using training datasets.

Training Data Set	Decision Tree Model	Classification Rate (%)	Ensemble Output (%)
A	M1	92.6	94.7
B	M2	87.9	91.8
C	M3	95.2	97.3

Table 5. Classification rate (%) of decision tree models using testing datasets.

Decision Tree Model	Classification Rate (%)
M1 using ID3	90.2
M2 using ID3	85.3
M3 using ID3	94.9
Proposed Ensemble Approach	98.4

5. CONCLUSIONS

The size of World Wide Web continues to grow at an amazing rate as the internet becomes globally popular. This paper introduced a new ensemble of decision tree classifiers that ensembles ID3 classifier for mining web data streams. ID3 classifier is very popular and useful data mining algorithms for supervised learning. We generated three training datasets with equal number of examples using web data streams, and built three decision tree models using each training dataset. For classifying the test or unseen examples: counts the weighted votes for each decision tree and assigns the class with the maximum weighted vote for that example. This ensemble approach can improve the classification rate of web data streams up to 98%. The future research issue will be web content mining and web structure mining.

6. ACKNOWLEDGMENTS

Support for this research received from the National Science & Information and Communication Technology (NSICT), Ministry of Science and Information & Communication Technology, Government of Bangladesh, and the Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh.

7. REFERENCES

[1] S. Ruggieri, "Efficient C4.5," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 2, 2002, pp. 438-444.

[2] Y. Zhao, and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Advances in Space Research*, Vol. 41, 2008, pp. 1955-1959.

[3] M. Hall, "A decision tree-based attribute weighting filter for naive Bays," *Knowledge-Based Systems*, Vol. 20, 2007, pp. 120-126.

[4] L. M. Wang, X. L. Li, C. H. Cao, and S. M. Yuan, "Combining decision tree and Naive Bayes for

classification," *Knowledge-Based Systems*, Vol. 19, 2006, pp. 511-515.

[5] D. V. Patil, and R. S. Bichkar, "An optimistic data mining approach for handling large data set using data partitioning technique," *International Journal of Computer Applications*, Vol. 24, No. 3, 2011, pp. 29-33.

[6] J. R. Quinlan, "Induction of Decision Tree," *Machine Learning*, Vol. 1, No. 1, 1986 pp. 81-106.

[7] J. R. Quinlan, "Decision trees and multi-valued attributes," In J.E. Hayes, D. Michie and J. Richards (eds.), *Machine Intelligence*, Vol. 11, 1988, pp. 305-318, Oxford, UK: Oxford University Press.

[8] L. Breiman, J. H. Friedman, R.A. Olshen, and C. J. Stone, "Classification and regression trees," *Statistics Probability Series*, Wadsworth, Belmont, 1984.

[9] J. R. Quinlan, "C4.5: Programs for machine learning," *Morgan Kaufmann Publishers*, San Mateo, California, 1993.

[10] S. K. Murthy, "Automatic construction of decision trees from data: A multidisciplinary survey," *Data and Knowledge Discovery*, Vol. 2, No. 4, 1998, pp. 345-389.

[11] D. Turney, "Cost-Sensitive Classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *Journal of Artificial Intelligence Research*, 1995, pp. 369-409.

[12] W. Z. Liu, and A. P. White, "The importance of attribute selection in decision tree induction," *Machine Learning*, Vol. 15, 1994, pp. 25-41.

[13] W. Buntine, and T. Niblett, "A further comparison of splitting rules for decision tree induction," *Machine Learning*, Vol. 8, 1992, pp. 75-85.

[14] S. R. Safavian, and D. Langrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 3, 1991, pp. 660-674.

[15] Dewan Md. Farid, Jerome Darmont, and Mohammad Zahidur Rahman, "Attribute Weighting with Adaptive NBTree for Reducing False Positives in Intrusion Detection," *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 8, No. 1, April 2010, pp. 19-26.

[16] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, "An Ensemble Approach to Classifier Construction based on Bootstrap Aggregation," *International Journal of Computer Applications (IJCA)*, Vol. 25, No. 5, July 2011, pp. 30-34.

[17] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, "Adaptive Intrusion Detection based on Boosting and Naïve Bayesian Classifier," *International Journal of Computer Applications (IJCA)*, Vol. 24, No. 3, June 2011, pp. 12-19.

[18] Dewan Md. Farid, Nouria Harbi, and Mohammad Zahidur Rahman, "Combining Naïve Bayes and Decision Tree for Adaptive Intrusion Detection," *International Journal of*



Network Security & Its Applications (IJNSA), Vol. 2, No. 2, April 2010, pp. 12-25.

- [19] Dewan Md. Farid, Jerome Darmon, Nouria Harbi, and Chowdhury Mofizur Rahman, "A New Supervised Learning Algorithm using Naïve Bayesian Classifier," In *Proc. of the IADIS International Conference Information Systems 2010*, 18-20 March, 2010, Porto, Portugal, pp. 78-84.
- [20] Rabindra Nath Das, Dewan Md. Farid, and Chowdhury Mofizur Rahman, "A New Decision Tree Learning Approach Using Bayesian Classifier," In *Proc. of the 3rd International Conference on Software, Knowledge, Information Management and Applications (SKIMA09)*, 21 - 23 October, 2009, Fes, Morocco, pp. 125-131.

8. AUTHORS PROFILE

Fauzia Yasmeen Tani is currently completing Master of Science in Computer Science and Engineering from Jahangirnagar University, Bangladesh. She obtained B.Sc (Hons.) in Computer Science and Engineering from the same institute in 2008. She is serving as a Lecturer, Department of Computer Science and Engineering, IBAIS University, Bangladesh. Her current research is focusing on Network Security and Data Mining.

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, and also a Guest Instructor in Post Graduate Diploma in ICT, Bangladesh Computer Council (Ministry of Science & ICT). He is a member of IEEE and IEEE Computer Society. He has published 6 international journals and 10 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. He received Senior Fellowship I & II awarded by National Science & Information and Communication Technology (NSICT), Ministry of Science & Information and Communication Technology, Govt. of Bangladesh, in 2008 and 2011 respectively.

Mohammad Zahidur Rahma is currently a Professor at Department of Computer Science and Engineering, Jahangirnagar University, Bangladesh. 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.