Journal of Theoretical and Applied Information Technology
15th July 2020. Vol.98. No 13
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE CLASSIFICATION IN DATA STREAM MINING

MOSTAFA M. YACOUB[1] , AMIRA REZK[2] , M. B. SENOUSY[3]

[1]Mr. Department of Information Systems, Faculty of Computers and Information, Mansoura University, Egypt

[2]Dr. Department of Information Systems, Faculty of Computers and Information, Mansoura University, Egypt

[3]Prof. Dr. Department of Computer and Information Systems, Faculty of Management Sciences, Sadat Academy for management Sciences, Egypt

E-mail: [1]mostafayacoub3@gmail.com, [2]amira_rezk@mans.edu.eg, [3]badr_senousy_arcoit@yahoo.com

ABSTRACT

Data streams gained obvious attention by researches for years. Mining this type of data generates challenges because of their special nature. Classification is one of the major approaches of Data Stream Mining (DSM). Concept drift (changes in pattern of data over time) is one of the major challenges that is needed to be adapted in data streams. Another challenge is high dimensional data streams.

This paper provides a review for classification techniques in adaptive data stream mining. Focusing on both challenges; concept drifts and dimensionality reduction and dividing these techniques into incremental and ensemble. Incremental classifiers such as Very Fast Decision Trees (VFDT) and Concept-adapting Very Fast Decision Trees (CVFDT) were tested. Adaptive Random Forests (ARF) was taken as an example for adaptive ensemble classifiers. Furthermore, a practical analysis between VFDT, CVFDT and ARF was held. The analysis was according to accuracy, processing speed, and tree size. Accuracy did not vary much between the three algorithms. ARF has much better results in speed and has the smallest number of tree nodes.

Keywords: *Data Stream Mining, Classification, Decision Trees, Adaptivity, Concept Drift*

1. INTRODUCTION

In today's information society, extraction of knowledge is becoming a very important task for many people. New emerging applications – in which data is made at very high rates in the form of passing data streams – became more popular by time. Examples of such applications include financial applications, network monitoring, security, telecommunication data management, web applications, manufacturing, sensor networks, and others.

The problems with data streams are that data is always on the move with high speed rates. Also, we only have one chance to gather these data and mine, because another flow is coming. This paper aims to experiment and analyze three of classification algorithms of data stream mining. This analysis aims to sort the three algorithms according to speed and accuracy. The lowest accuracy algorithm and the highest time will be subject to modifications and enhancements in future work. The analysis was done on the same machine with same datasets and unified attributes.

Second section demonstrate related work to that research and third one describes briefly popular classification methods classified into two main groups; incremental and ensemble. Fourth section gives an overview of dimensionality reduction and adaptive random forests, two methods for adaptive DSM. Fourth section gives an experimental study, in which we compared three algorithms according to accuracy, time, and tree size.

2. RELATED WORKS

VFDT and CVFDT are two of the most popular classification algorithms in data stream mining. They have been used in analysis and modification trials many more times. Nishimura [1] held and comparison of C4.5, CVFDT and CVFDT with Naïve-Bayes Classifiers "$CVFDT_{NBC}$". Performance test was with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

artificial data streams with concept drift. In time measures, $CVFDT_{NBC}$ has mush bigger time than CVFDT but with smaller tree size. Priyadarshini [2] compared VFDT and CVFDT along with weighted ensemble and multi ensemble. Comparison was according to time complexity, space complexity and concept drift handling. VFDT shoes less space and time complexity and proved to handle concept drifts. Vivekanandan [3] compared CVFDT with Ensemble based classifier (EC), Rule Based Classifier (RBC), and Genetic algorithm (GA). CVFDT showed higher error rate than other algorithms. Kosina [4] proposed six variations of very fast decision rules (VFDR), which is an extension of VFDT. Parita [5] demonstrated a review on VFDT and CVFDT, explaining in details the steps and attributes of each algorithm. Li [6] proposed EDTC (Ensemble Decision Trees for Concept drifting data streams). Then compared the algorithm with original VFDT and CVFDT among other variations of them. Frías-Blanco [7] compared VFDT with four other algorithms. Comparison was based on accuracy, time, and number of nodes. VFDT showed the best time and third best accuracy.

3. DATA STREAM CLASSIFICATION

Data Streams have unique features that add more complications when dealing with this type of data. These features include continuous flow, infinity, high speed arrival, storage disability, one-time processing, large volume, and subject to change over time.

Classification is finding a model from existing data that can be applied to upcoming data. That model will be able to predict a class for these future data. In other words, classification is to discover the common features of group of data objects and classify them into different classes according to the classification pattern [8].

Learning algorithms can be divided into four main categories according to time constraints and example availability. These types are Batch mode learning algorithms, incremental algorithms, on-line learning, and any-time learning [9].

In batch learning, the algorithm processes small volumes of data which is available before starting the learning process.

Another categorizing of data stream classification methods is incremental and ensemble methods [10]. In incremental learning, a single classifier is being used and updated incrementally to deal with new class labels. Incremental methods include decision trees, Support Vector Machines "SVM", and Bayesian.

However, ensemble methods combine multiple classifiers to achieve higher accuracy. Which means that ensemble classifiers gain better predictive performance than incremental classifiers.

3.1 Incremental Classifiers

In this section, two types of incremental classifiers were presented. Decision trees and support vector machines are visited comprehensively followed by detailed review of VFDT and CVFDT algorithms.

3.1.1 Decision Trees:

Decision trees are hierarchical data structures for supervised learning by which the input space is split into local regions to predict the dependent variable [11].

Decision trees consist of nodes, branches, and leaves. The root is the first node of the tree. The root and nodes carry tests over a feature. Branches are the links between nodes according to answers about those tests in nodes. Each branch represents a value. Finally, leaves are the last nodes of the tree and have no tests or branches coming out of it. Each leaf assigns a class.

The best attribute to split is quantified based on an impurity measure. Impurity measures include information gain or Gini index.

Information Gain is an impurity based criterion that uses entropy measures as the impurity measure [12][13]. Where Gini Index measures how often an element would be incorrectly classified randomly, according to the distribution of classes in the subset [14]. In other words, it measures the level of impurity or inequality of the samples assigned to a node based on a split at its parent [15].

Decision trees are used widely as they are simple, easy to represent and understand, more accurate, and greedy recursive induction; as they maximize the information gain at each node. In next section, we present two of the most used decision tree algorithms; Very Fast Decision Trees (VFDT) and Concept-adapting Very Fast Decision Trees (CVFDT).

Very Fast Decision Trees (VFDT):

VFDT was presented in 2000 [16]. In VFDT, a decision tree is inducted by recursively replacing leaves with decision nodes. Each leaf stores sufficient statistics about attribute-values. Those

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

statistics are needed by a heuristic function that evaluates the quality of split-tests based on attribute-values [17]. The tree is split by using the current best attribute taking into consideration that the number of examples used satisfies the Hoeffding bound [18]. The algorithm have three main steps; calculate quality measure for each attribute, use Hoeffding bound to decide if existing data is sufficient to be split, and if data is sufficient then it make the split and create subnodes which will be split next.

According to Hoeffding bounds, n independent observations of a real-valued random variable r with range R, with confidence 1 – δ, the true mean of r is at least $\bar{r}$ - ε, where $\bar{r}$ is the observed mean of the samples and

$$\varepsilon = \sqrt{\frac{R^2 ln\frac{1}{\delta}}{2n}} \quad [19] (1)$$

Because of dealing with data streams, VFDT has no termination point and the tree keeps growing. To minimize the excessive use of limited memory, the algorithm can exclude inactive leaves and remove poor attributes with their associated statistics.

Concept-adapting Very Fast Decision Trees (CVFDT):

CVFDT was predicated from VFDT to deal with concept drift which is an important consideration when dealing with stream data [20].

Concept drifts are possible changes in the concept or the functional dependencies being modeled. In other words, concept drift occurs when the class distribution changes over time. These drifts demands detection from streaming algorithm, and adapting the model [17] [21] [22] [23]. Weather forecasting, spam recognition, and monitoring systems are examples in which concept drifting is a challenge [10].

CVFDT maintains VFDT's speed and accuracy but adds the ability to detect and handle changes. It keeps the model consistent with fixed-size sliding window of recent examples. The algorithm uses statistics stored at each node about the current data to detect concept drifts.

When the concept is changing and a split that was previously selected would no longer be the best, the algorithm starts learning an alternate subtree with a new best attribute as its root. Which means the algorithm does not start a model from scratch. When the subtree become more accurate, it replaces the original one. The algorithm keeps the model up-to-date when there are large and frequent changes in the concept [4].

3.1.2 Support Vector Machines (SVM):

Classification is done in Support Vector Machines (SVM) by computing a hyperplane in the vector space that separate positive and negative instances and maximize the distances from the hyperplane to the closest positive and negative examples [24]. In other words, it's main idea is to maximize the distance between the separating hyperplane and the closest training examples [9].

SVM can be either linear on non-linear kernels. It uses Kernel function to classify data into higher dimension then find the best hyperplane that separates the patterns between classes. The best hyperplane is the one with the maximum margin between classes. The support vectors are the data points that are closest to the separating hyperplane [25]. Kernel selection, kernel's parameters, and the soft margin parameter C are the effectiveness factors of SVM [26]. SVM is useful in handling non-linear, sparse, noisy, and high-dimensional problems. One major disadvantage of SVM is computational cost which has been subject to optimizations to increase its scalability [27] [28].

3.2 Ensemble Classifiers:

As mentioned earlier, ensemble methods utilize multiple learning algorithms to gain better predictive performance than single classifiers. This performance advantage is a result of that single classifier may not learn all different distribution of patterns. Better performance is not the only purpose of ensemble classifiers, processing speed and response time are two additional benefits of ensemble classifiers. Two most common ensemble types are averaging and boosting.

In averaging methods, several learners are run independently, and their predictions are averaged. Two most common averaging methods are bagging and random forest. In boosting methods, weak learners are ordered according to error rates [29].

Bagging is based on random sampling, while boosting works by windowing. Compared with boosting, bagging is more noise tolerant and produced better class probability estimates.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Ensemble learners can be grouped into four main categories according to combination, diversity, base learner, and update dynamics [30]. Another classification for ensemble methods is found in [25], in which ensemble classifiers are divided into six groups according to what they are based on.

4. ADAPTIVE CLASSIFICATION:

There are different ways to adapt classification in data stream mining. In this section we provide a comprehensive review of two adaptive techniques; dimensionality reduction and adaptive random forests.

4.1 Dimensionality Reduction:

As data has more dimensions, data mining algorithm become more computationally difficult and inapplicable in many real applications. In high dimensional data, similarities between points decreases. So, the expected gap between the Euclidean distance to the closest neighbor and that to the farthest point decreases as dimensionality grows. This makes data mining algorithms inapplicable and vulnerable to noise. In dealing with high dimensional data, data mining algorithm should be fast, scalable, and perform dimensional reduction.

Dimensionality reduction is "the process of taking data in a high dimensional space and mapping it into a new space whose dimensionality is better" [31]. Dimensionality reduction means transforming dataset X with dimensionality D into a new dataset Y with dimensionality d, without changing the structure of the data as much as possible.

Missing value ratio is one of dimensionality reduction techniques. In which, data columns with a lot of missing values will not be much useful and can be removed. Reduction can be more aggressive by making threshold higher. Another similar technique is low variance filter. Data with little changes (variance) can be omitted.

In High correlation filter technique, columns with very similar trends – carry very similar data – can be shorted in only one column. Random forests are another technique, which will be discussed later in this section.

Principal Component Analysis (PCA) is one of common dimensionality reduction techniques. Principal components are orthogonal directions that capture most of the variance in the data. PCA performs reduction by embedding the data into a linear subspace of lower dimensionality. PCA has been successfully applied in many areas such as; face recognition, coin classification, and seismic series analysis [32]. Although PCA reduces complexity of data and identify most important features, it could ignore useful information [33]. In another classification of dimensionality reduction, techniques falls in two main groups; convex techniques and non-convex techniques [32].

4.2 Adaptive Random Forest

The main difference between random forests and other trees is randomness. This randomness is used in splitting nodes. In regular trees, split is done by choosing the best split among attributes. In random forests, random predictors are chosen to split a node. Which means that every tree is grown based on a different random sample of data. After generating multiple models, they are combined by voting.

Random forests offer data visualization of high dimensional data and error detection. In addition to, high accuracy and resistance to overfitting. Ransom forest has the ability to model high dimensional data as it can handle missing values, solve the problem of over-fitting, and no need for tree pruning [34].

5. EXPERIMENT STUDY:

In this paper, an experiment was done on VFDT, CVFDT, and ARF using Massive Online Analytics (MOA) working on Microsoft Windows 8 environment on core i5-5200U processor machine. Visualization and charts were done using Microsoft PowerBI. Airlines data set [35] is a classification dataset with 20 attributes and nearly 539,000 records. Also, KDD99 dataset has been used in figures 4 and 5. KDD99 is a multivariate dataset with 42 attributes and 400000 instances.

Comparison has been held according to three criteria. Classification correct percentage is the percentage of the accuracy of the algorithm. Evaluation time is the time required by the algorithm to process the whole sample. Tree size is the number of nodes the algorithm needed to reach the leaves that we have mentioned earlier.

Figure 1 displays a comparison between the three algorithms according to classification correct percentage. Differences are not significant between the three algorithms. However, minimum accuracy recorded was %44.8 with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ARF. VFDT and CVFDT minimum accuracy was %50.1 and %53.9 respectively.

Average accuracy for ARF, CVFDT, and VFDT was %61.5, %63.8, and %65 respectively. Highest accuracy was % 84.3 for both ARF and CVFDT.

Figure 2 demonstrate the evaluation time for the three algorithms. ARF has a significant lower time than the other two. Total time was 40.4 seconds. While CVFDT was 60 seconds and VFDT was the highest time with 68 seconds.

In figure 3, the tree size is displayed with high differences between the three algorithms. VFDT has the lowest tree size with 8582 nodes. CVFDT was the highest number of nodes, 91376 nodes. 34651 nodes were the total tree size of ARF algorithm.

Figures 4 and 5 are used to represent KDD99 dataset according to classification correct percentage and evaluation time. In figure 4, accuracy has reached %100 with the three algorithms at most of the evaluation time. However, the minimum accuracy was for VFDT algorithm with %73.1, while it was %80 at both CVFDT and ARF.

Figure 5 displays the evaluation time. VFDT has finished the evaluation in 33.3 minutes, while it reached 8.5 minutes with CVFDT. ARF was the shortest time – with small difference with CVFDT – with 8 minutes.

6. CONCLUSION AND FUTURE WORK:

In this paper, we reviewed different mechanisms of classification in data stream mining, with focus on both incremental and ensemble classifiers. In adaptive classification we presented dimensionality reduction as a well-known adaptive technique in data mining in general, not just data stream mining. ARF was presented as a next step for adaptive classification after decision trees (CVFDT).

In experimental part, we compared between ARF, CVFDT, and VFDT. With instances keep coming, the three algorithms nearly the same classification correction percentage. As for processing speed, ARF clearly get the lowest processing time, nearly half the time. VFDT showed the longest time of the three algorithms. CVFDT generates much more nodes than VFDT and ARF. ARF has the lowest number of nodes with noticeable difference.

In future work, we are working on modifications for both CVFDT and ARF. CVFDT will be modified to reduce the evaluation time. ARF also is subject to enhancements in time and accuracy. Accuracy can be improved by training each tree separately from other trees with taking time in consideration. VFDT should be enhanced according to time measure.

REFERENCES:

[1] S. Nishimura, M. Terabe, K. Hashimoto, and K. Mihara, “Learning higher accuracy decision trees from concept drifting data streams,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5027 LNAI, pp. 179–188, 2008.

[2] Priyadarshini.I.Kuchanur, “A Survey on New Techniques Over Data Streams,” Int. J. Innov. Res. Comput. Commun. Eng., vol. 7, no. 3, pp. 3056–3063, 2016.

[3] P. Vivekanandan and R. Nedunchezhian, “Mining data streams with concept drifts using genetic algorithm,” Artif. Intell. Rev., vol. 36, no. 3, pp. 163–178, 2011.

[4] P. Kosina and J. Gama, “Very fast decision rules for classification in data streams,” Data Min. Knowl. Discov., vol. 29, no. 1, pp. 168–202, 2013.

[5] P. Parita and P. Singh, “A Review on Tree Based Incremental Classification,” Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 1, pp. 306–309, 2014.

[6] P. Li, X. Wu, X. Hu, and H. Wang, “Learning concept-drifting data streams with random ensemble decision trees,” Neurocomputing, vol. 166, pp. 68–83, 2015.

[7] I. Frías-Blanco, J. Del Campo-Ávila, G. Ramos-Jiménez, A. C. P. L. F. Carvalho, A. Ortiz-Díaz, and R. Morales-Bueno, “Online adaptive decision trees based on concentration inequalities,” Knowledge-Based Syst., vol. 104, pp. 179–194, 2016.

[8] Y. Hou, “Decision Tree Algorithm for Big Data Analysis,” Adv. Intell. Syst. Res., vol. 161, no. Tlicsc, pp. 270–274, 2018.

[9] V. Lemaire, C. Salperwyck, and A. Bondu, “A survey on supervised classification on data streams,” Bus. Intell., pp. 88–125, 2015.

[10] S. Homayoun and M. Ahmadzadeh, “A review on data stream classification approaches,” J. Adv. Comput. Sci. Technol., vol. 5, no. 1, p. 8, 2016.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[11] E. Alpaydın, "Introduction to machine learning," Methods Mol. Biol., vol. 1107, pp. 105–128, Apr. 2014.

[12] K. J. Cios, W. Pedrycz, and R. W. Swiniarski, "Data Mining and Knowledge Discovery," in Data Mining Methods for Knowledge Discovery, Boston, MA: Springer US, 1998, pp. 1–26.

[13] L. Rokach and O. Maimon, "Decision Trees," Data Min. Knowl. Discov. Handb., pp. 165–192, 2010.

[14] C. Diamantini, L. Genga, D. Potena, and E. Storti, "New Frontiers in Mining Complex Patterns," Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci., vol. 8983, no. Nfmcp, pp. 149–163, 2015.

[15] R. Polikar, Ensemble Machine Learning. Boston, MA: Springer US, 2012.

[16] P. Domingos and G. Hulten, "Mining high-speed data streams," Proc. sixth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '00, pp. 71–80, 2000.

[17] J. Gama, Knowledge Discovery from Data Streams. CRC Press, 2010.

[18] M. M. Gaber, "Advances in data stream mining," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 2, no. 1, pp. 79–85, 2012.

[19] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," J. Big Data, vol. 1, no. 1, pp. 1–16, 2014.

[20] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Second. Elsevier Inc., 2006.

[21] L. Rokach, "Ensemble-based classifiers," Artif. Intell. Rev., vol. 33, no. 1–2, pp. 1–39, 2010.

[22] E. Ikonomovska, "Algorithms for Learning Regression Trees and Ensembles on Evolving Data Streams," Thesis - PHD, p. 222, 2012.

[23] J. A. Gama, "A Survey on Concept Drift Adaptation," ACM Comput. Surv, vol. 1, no. 35, 2013.

[24] C. K. S. Leung and R. K. MacKinnon, Balancing tree size and accuracy in fast mining of uncertain frequent patterns, vol. 9263. 2015.

[25] A. Rahman and S. Tasnim, "Ensemble Classifiers and Their Applications: A Review," Int. J. Comput. Trends Technol., vol. 10, no. 1, pp. 31–35, 2014.

[26] D. Hutchison, Machine Learning in Medical Imaging, vol. 10019. 2016.

[27] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT)," Procedia Comput. Sci., vol. 58, no. DaMIS, pp. 437–442, 2016.

[28] G. Williams, Data Mining with Rattle and R. 2011.

[29] D. R. Tobergte and S. Curtis, Designing Machine Learning Systems with Python, vol. 53, no. 9. 2016.

[30] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A Survey on Ensemble Learning for Data Stream Classification," ACM Comput. Surv., vol. 50, no. 2, pp. 1–36, 2017.

[31] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory to algorithms, vol. 9781107057. 2013.

[32] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, "Dimensionality Reduction: A Comparative Review," J. Mach. Learn. Res., vol. 10, pp. 1–41, 2009.

[33] C. Lam, Machine Learning in Action. 2011.

[34] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," IJCSI Int. J. Comput. Sci. Issues, vol. 9, no. 5, pp. 272–278, 2012.

[35] E. Ikonomovska, "Airline," 2009. [Online]. Available: https://kt.ijs.si/elena_ikonomovska/data.html. [Accessed: 01-Aug-2019].

Journal of Theoretical and Applied Information Technology
15th July 2020. Vol.98. No 13
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

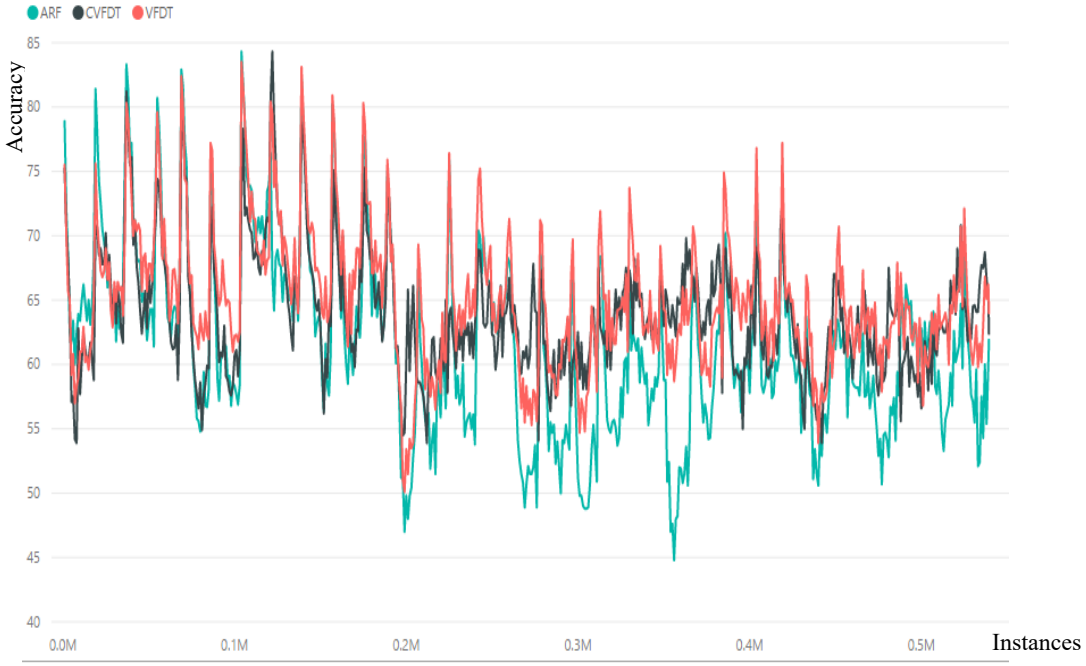

Figure 1: Classification Correct Percentage

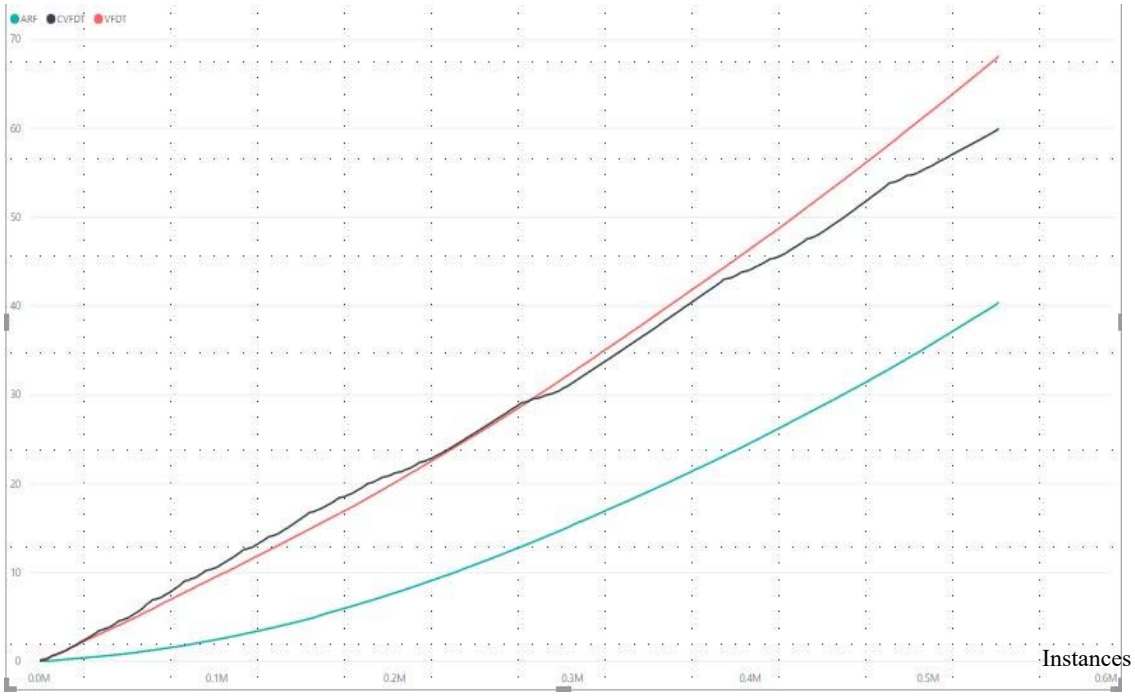

Figure 2: Evaluation Time (CPU seconds)

Journal of Theoretical and Applied Information Technology
15th July 2020. Vol.98. No 13
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

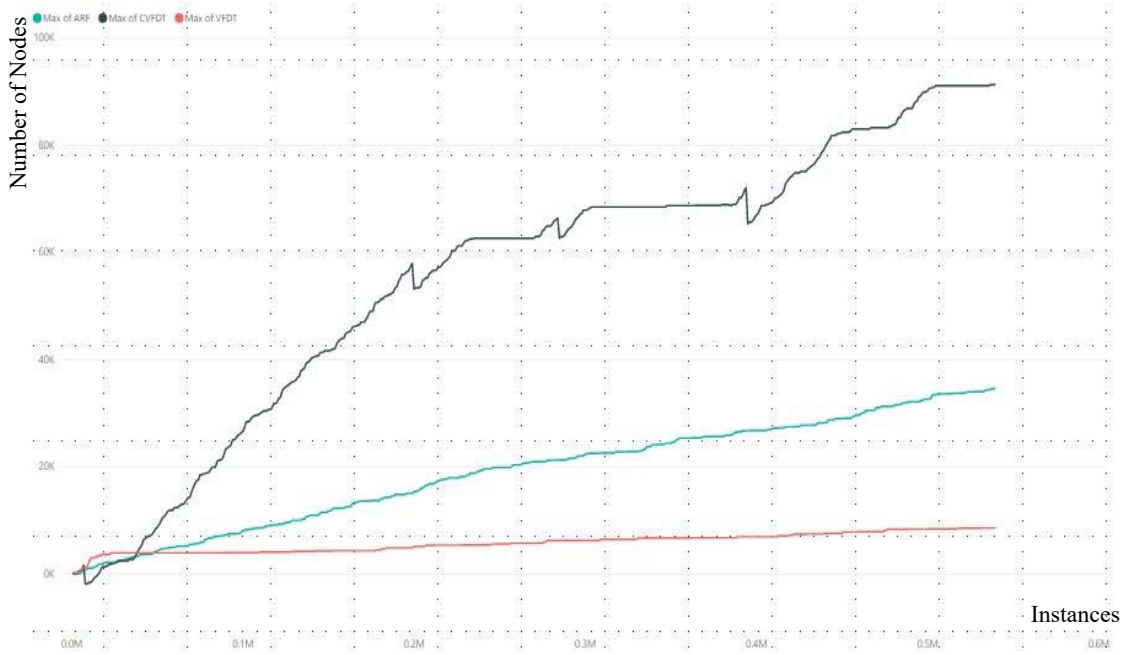

Figure 3: Tree Size (nodes)

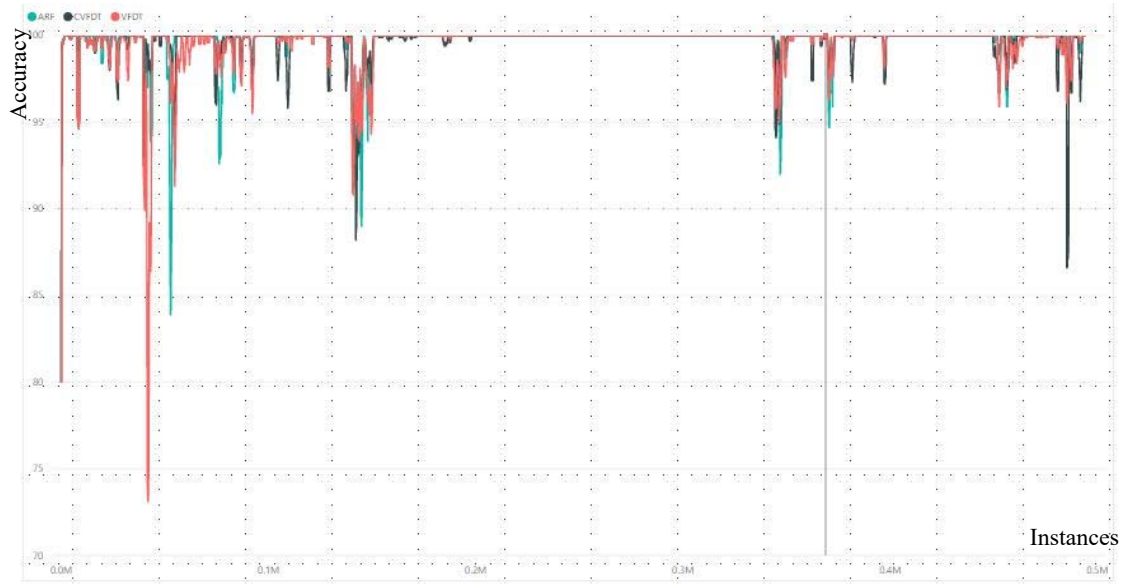

Figure 4: Classification Correct Percentage

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

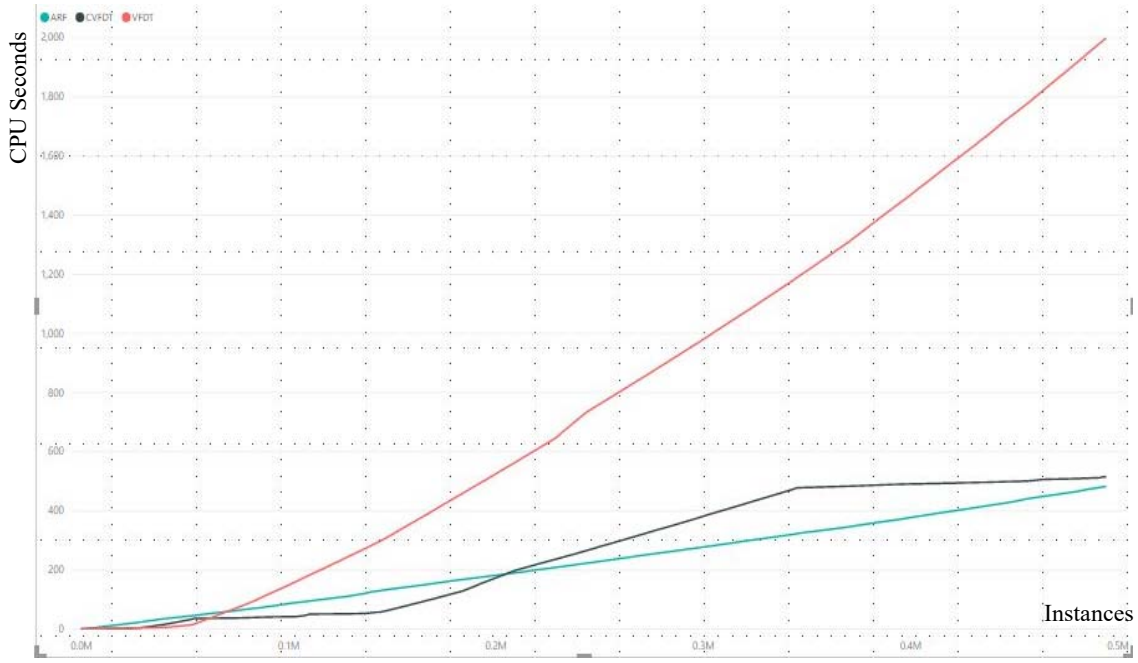

Figure 5: Evaluation Time