International Journal of Innovation and Scientific Research
ISSN 2351-8014 Vol. 32 No. 2 Sep. 2017, pp. 345-353

http://www.ijisr.issr-journals.org/

AN IMPROVEMENT CLASSIFICATION ALGORITHM UTILIZING STREAMING DATA

Hind Ra'ad Ibraheem and Enas Mohammed Hussein

Computer Science Department, AL-Mustansiriyah University, Iraq

ABSTRACT: The data stream has recently emerged in response to the continuous data problem. Stream data is usually in vast volume, changing dynamically, possibly infinite, and containing multi-dimensional features. The attention towards data stream mining is increasing as regards to its presence in wide range of real-world applications, such as e-commerce, banking, sensor data and telecommunication records. Similar to data mining, data stream mining includes classification, clustering, frequent pattern mining etc. techniques; the special focus of this paper is on classification methods invented to handle data streams. Performance of data stream classification is measuring by involving processing speed, memory and accuracy. Also, a classification algorithm must meet several requirements in order to work with the assumptions and be suitable for learning from data streams so studying purely theoretical advantages of algorithms is certainly useful and enables new developments. Here we present a comprehensive survey of the state-of-the-art data stream mining algorithms with a focus on classification because of its ubiquitous usage. It identifies mining constraints, proposes a general model for data stream mining, and depicts the relationship between traditional data mining and data stream mining. In this paper, we propose a new streaming data classification algorithm based on Hoeffding tree algorithm called Fast Decision Tree Algorithm (FDTA) as an improvement method to classify stream data and compared between them according to the three measures which are classification accuracy, memory space and execution time.

KEYWORDS: Data Stream, Data Stream mining, Data Stream classification, Hoeffding tree algorithm, FDTA.

1 INTRODUCTION

Dramatic growth in information technology and vast volume of generated data has made new challenging discovery tasks in processing of data. The term "data stream" is defined as a sequence of data that arrives at a system in a continuous and changing manner. Data stream can be conceived as a continuous and changing sequence of data that continuously arrives at a system to be stored or processed [1].

Data streams have some characteristics in common such as massive, temporally ordered, fast-changing and potentially infinite in length. According to [2], there are some reasons which distinguish data streams from traditional data mining:

- The size of data streams is potentially boundless.
- The elements of stream arrive on-line.
- Because of limitations in memory space, after processing of an element, system discards (or summarizes) it.
- The system cannot control or determine how data elements arrive.

Emails, sensor data, websites customer click stream, network traffic, weather forecasting data etc. are some examples of data stream. Data stream mining comprises three main techniques such as clustering, classification and frequent pattern mining.

Other sources can be sensors situated in medical domain to monitor health conditions of patients, twitter posts and many more [3]. The above mentioned sources are not only produce stream data, but they produce them in huge amount (of scale in tear bytes or peat bytes) and at high speed. Now mining such huge data in real time raises various challenges and has become a hot area of research recently. These challenges include memory limitation, faster computing requirements etc. Apart of these

Corresponding Author: Hind Ra'ad Ibraheem

challenges, streaming data has inherent the evolution that means the concept being mined evolve and change over time. This challenge itself poses several other issues in streaming data mining [4].

The data stream mining task can be consider same as traditional data mining task in terms of objective but quite different in terms of processing or the executing mining task. The reason behind this difference is the underlying of infinite high speed data streams. It makes the traditional data mining algorithms and techniques incapable of appropriately handling data stream and yields the requirement of algorithm suitable for streaming data mining [5].

This paper presents an overview of streaming data mining along with major issues and challenges associated with it. Section-1 introduces the data stream as well as the need of stream data mining, new algorithms, performance measures. Section-2 describe algorithms of classification available for streaming data mining. Then after section-3

2 STREAM DATA MINING ALGORITHMS

Data stream mining are classified basically in classification, clustering and pattern mining. One goal of data stream mining is to create a learning process that linearly increases according to the number of examples. Moreover, as data continuously arrive with new information, the model that was previously induced not only needs to incorporate new information, but also eliminates the effects of outdated data. Simply retraining the model with new examples is ineffective and inadequate; therefore, another goal of data stream mining is to update its model incrementally as each example arrives [6]. The focus of this paper is on classification techniques. In stream data mining scenario, following significant algorithms in category of classification are available:

2.1 STREAM DATA CLASSIFICATION ALGORITHMS

Classification is a supervised learning techniques which aims to predict of an independent variable (class label) according to some values of an instance. Making a classification model has two main phases: 1) Model creation, 2) Model evaluation. At the first phase, a learning algorithm uses dataset to create a model which is able to predict class label. The second phase tries to investigate the accuracy parameters of created model [7].

Various classification algorithms for streaming data have been devised from time to time in the last decade. Each algorithm has its own capabilities and key focus to avert challenges of stream data mining. Some of available streaming data classification algorithms along with their key features are chronologically listed in table (1.1)

Streaming classification algorithm	Year	Key features
VFDT[8]	2000	It uses Hoeffding bound to assess the number of minimum instances demand to grow decision tree. Require lesser memory.
CVFDT[9]	2001	It is the advancement of VFDT that has the ability to adaptation to concept drift.
Streaming ensemble algorithm (SEA)[10]	2001	Provide robustness and treats concept drift but need to be carefully used with high speed data stream.
UFFT [11].	2004	Introduced a forest having binary trees (for each pair of class) in case of multiclass problem. It utilize restricted memory.
IOLIN [12].	2008	Variation of OLIN that preserve on model updating until sufficient concept drift thereby keeps computational effort significantly.
Similarity-based Data Stream Classifier [13].	2013	Uses a new insertion/removal methods for quickly capturing and representing changes in data to enhance performance. Also incorporate new class labels and discards obsolete class labels during the execution.

2.2 HOEFFDING TREE ALGORITHM

Before starting a Hoeffding algorithm, first of all we define classification problem is a set of training examples of the form (x, y), where x is a vector of d attributes and y is a discrete class label. Our goal is to produce from the examples a model y=f(x) that predict the classes y for future examples x with high accuracy. Decision tree learning is one of the most effective

classification methods. A Decision tree is learned by recursively replacing leaves by test nodes, starting at the root. Each node contains a test on the attributes and each branch from a node corresponds to a possible outcome of the test and each leaf contains a class prediction. All training data stored in main memory so it's expensive to repeatedly read from disk when learning complex trees so our goal is design decision tree learners than read each example at most once, and use a small constant time to process it[14].

So key observation is find the best attribute at a node. So for that consider only small subset of training examples that pass through that nodes. Choose the root attribute. Then expensive examples are passed down to the corresponding leaves, and used to choose the attribute there, and so on recursively. So use Hoeffding bound to decide, how many examples are enough at each node???

$$\varepsilon = \sqrt{(R2 \text{ in } (1/\delta)/2n} \quad \ldots \quad (1)$$

Algorithm (1.1) Hoeffding tree induction algorithm.

1: *HT* be a tree with a single leaf (the root)

2: **for all** training examples **do**

3: Sort example into leaf *l* using *HT*

4: Update sufficient statistics in *l*

5: Increment *nil,* the number of examples seen at *l*

6: **if** *NL mod Nmin* = 0 **and** examples seen at *l* not all of same Class **then**

7: Compute l (*XL*) for each attribute

8: Let *Ax* be attribute with highest l

9: Let *Xb* be attribute with second-highest l

10: Compute Hoeffding bound

11: **if** $Xa \neq X\emptyset$; **and** (l (*Xa*) - l (*Xb*) > ε **or** ε < *τ)* **then**

12: Replace *l* with an internal node that splits on *Xa*

13: **for all** branches of the split **do**

14: Add a new leaf with initialized sufficient statistics

15: **end for**

16: **end if**

17: **end if**

18: **end for [15].**

A. Strengths and weakness of Hoeffding tree algorithm

It contains such advantage, in that one is Scales better than traditional methods in terms of Sub linear with sampling and it utilization very small memory. Second it makes class predictions in parallel and new examples are added as they come.

Also, Hoeffding tree algorithm have Weakness that one is could spend a lot of time with ties. Second is Memory used with tree expansion and large Number of candidate? Attributes [16].

3 THE PROPOSED ALGORITHM

Our proposal will called ***fast decision tree (FDT)*** which is an improvement on the basic algorithm of Hoeffding tree, which is suffering from decided between two very similar attribute(when two attributes has identical gain) it takes long time to decide between them, *t* (*tie breaking*) is an algorithm is used to decide between them. *T* has a domain and fall in (0-1), in experiments is it specified as 0.05. From the experiment *t* also shown has the effect on the splitting process in order to create the decision tree. Accordingly, *t* will generated in many values by experiments it had reached to the best values between (0-1) increasing in order fashion, and with each generated value the three standard measurements will be computed which are the

classification accuracy, memory space and execution time, and return back the value which presents the highest classification accuracy. As will present in the below algorithm (1.2). Also Figure **1** below will illustrate general view of the proposal system.

Algorithm (1.2): Fast Decision Tree Algorithm (FDTA).

Input: N*min*, $^{\delta}$, N*l* = 0, *t* =0.0, ink-value, max-inc and a set of training examples.
Output: Decision tree, classification accuracy, memory space and execution time.

Process

1. Start a tree with a single leaf node *f* (the root).
2. **For** all training examples **do**

Begin

2.2.1 Using Hoeffding tree (*HT*) to sort the examples.
2.2.2 Modify the sufficient statistics in *f*
2.2.3 Increment N*L*.
2.2.4 **If** N*l* **mod** N*min*=0 **and** examples seen not regard to the same class **then**

Begin

a. Calculate the gain G for each attribute (Yi).
b. Pick out the two highest gain values as Ya and Yb.
c. Let Y0 = the attribute with less gain value(NULL).
d. Compute R = Ya- Yb.
e. Calculate HB (€) using equation (2.9).
f. t=t+inc_value
g. if t ≠ max_inc then

begin

If Ya<> Y0 and (R > € or € <*t*)
then
Displace *f* with an interior node that incises on Ya.
Else return to step f.
For all branches of the incise do
Create a new leaf with initialized sufficient statistics.
End For
Compute classification accuracy, execution time and memory space
Else return to step f
End IF
Else return to step 2.2.3
End IF

2.2.5 **Return** *t*-value with the highest accuracy ,FDT, execution time & memory space

End For
End Process

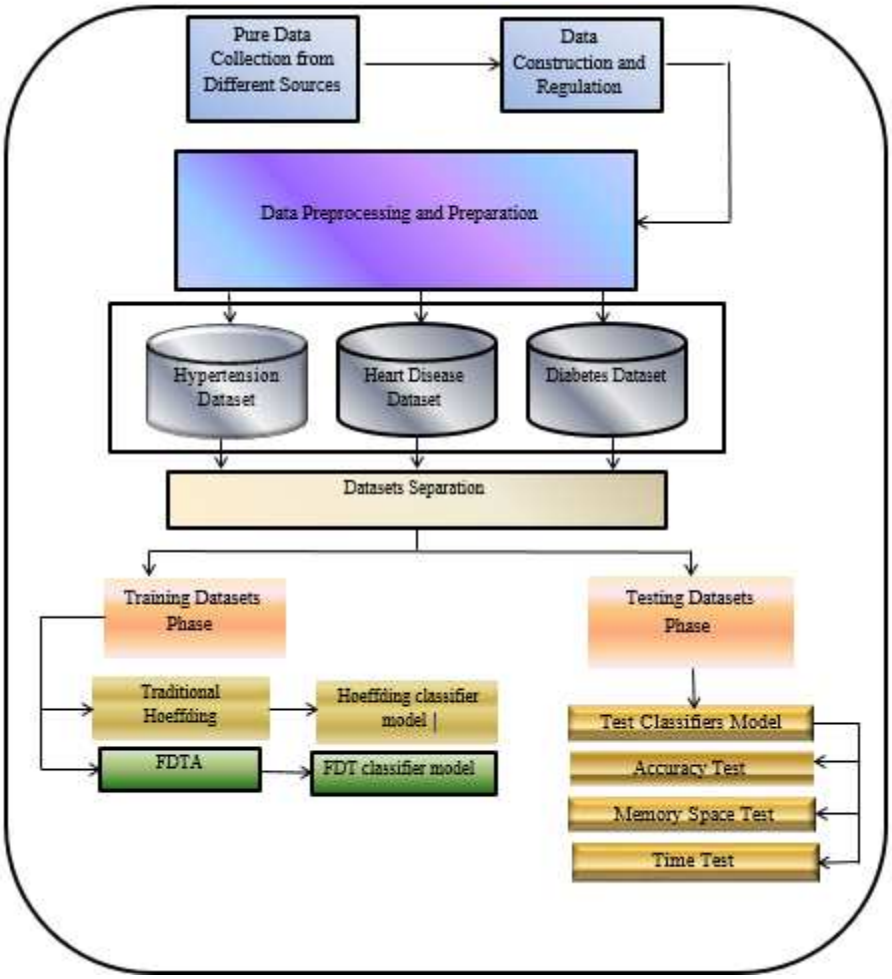

Figure 1 general view diagram

The above diagram will explained in the following steps:

1- **Pure Data Collection from Different Sources:** datasets were collected from different sources.as shown in the above diagram there are three datasets were used. The first dataset are hypertension dataset were gathered from real world (from Iraqi hospital), whereas data were collected directly from the patients file. And many preprocessing steps conducted on it in order to preparing it and be convenient to applying classification algorithms. The two others datasets which are diabetes and heart disease datasets were taken from *UCI* machine learning repository.

2- **Data Construction and Regulation:** Here these collections of diverse data were constructed analyzed. Each data disease was organized according to the set of disease indicators and its possible values.

3- **Data preprocessing and preparing:** Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results. In order to improve the quality of data, consequently, improve the quality mining results, theses data must be processed. It is the most important step, whereas after data have been collected from various sources, now it must be cleaned from a noisy, outliers and missing values (*data cleaning*). *Data integration* merges data from multiple sources into a cohesive data store such as a data warehouse. *Data reduction* can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering. Removing redundancies and normalization another significant stages in preprocessing. Feature selection also another significant stage in preprocessing to choose the most important features only as an active indicator to classify instances. The three datasets were cleaned and removed any redundancies and missing values. And the attributes value be normalized into a specific range in order to facilitate instances classification process.

4- **Datasets Separation**: After the datasets had been processed, a simple splitting partitions are conducted to the datasets that divided it into two subsets which are the training set (which the algorithm is applied on it to build the classifiers models)

and the second one are the testing subset (these subsets used to evaluated the induced classifier).It is common to designate (2/3) of the data as training data sets and (1/3) of the data as testing data sets. This stage includes two phases:-

- *Training phase*: involves building a classifiers models induced by applying above mentioned four classification algorithms.
- *Testing phase*: the evaluation of the induced classifiers in this step.

The evaluation being under different measurements which are calculated through classification accuracy, memory space, execution time, error rate and precision.

5- Applying the classification: traditional Hoeffding tree algorithm and fast decision tree algorithm also are carrying out on the training datasets and a classifier models were created depending on it.

After that the testing phase is beginning in order to check the classifiers validity. The classifiers are tested according to the classification accuracy, memory space and execution time.

4 DATASET USED

Dataset is a collection of set of information which is comprised of separate elements**.** Here we gathered the three datasets from different sources, the first dataset which are hypertensive dataset, we obtained it directly from the Iraqi hospitals(collected manually), the patients file were taken and data which include medical information about their health like analyses laboratory results and hypertension measurement were extracted and be analyzed and preprocessed. The second and third datasets are heart disease and diabetes dataset are obtained from *UCI* repositories. The datasets then also preprocessed to be ready to apply the two algorithms. We focused on an important attributes like cholesterol, serum triglyceride, serum low density lipoprotein, serum high density lipoprotein, serum very low density lipoprotein. Which are considered a very significant indicator to decide if the person is sick or healthy. There are four various datasets sizes in our datasets (10000-25000-50000-100000) instances. This dataset used for classifying each instance as sick or healthy.

5 EXPERIMENTAL RESULTS

We developed an application in visual basic language**,** the application is carried out with several steps**.** So accordingly, evaluation of results will be shown and accuracy, time (by time we mean the time needed to build a tree) and memory space has been considered for measuring the overall performance. The three datasets were used in order to build the two classifier. The first one induced from applying Hoeffding tree algorithm and the second one was induced from the proposed algorithm (FDTA). Then these classifiers were tested and compared according to their obtained result.

6 ACCURACY

It means that how much our system is accurate enough to classify between normal and anomalous behavior. It is calculated as,

$$A = TN + TP \mid TP + TN + FP + FN \quad (2)$$

Where,

- TP – is the True Positives which mean positive cases are correctly identified.
- TN – is the True Negatives which mean positive cases are incorrectly identified.
- FP – is the False Positives which mean negative cases are incorrectly identified as positive.
- FN – is the False Negatives which mean positive cases are incorrectly identified as negative.

Table 1.2 Hoeffding vs. FDTA for Data set-1

Data set Size	Training records number	Tasting records number	Hoeffding Accuracy (%)	FDTA Accuracy (%)
10000	6667	3333	47.7047	72.7668
25000	16667	8333	48.3739	71.6260
50000	33333	16667	49.1690	73.5929
100000	66676	33334	50.2344	72.8443

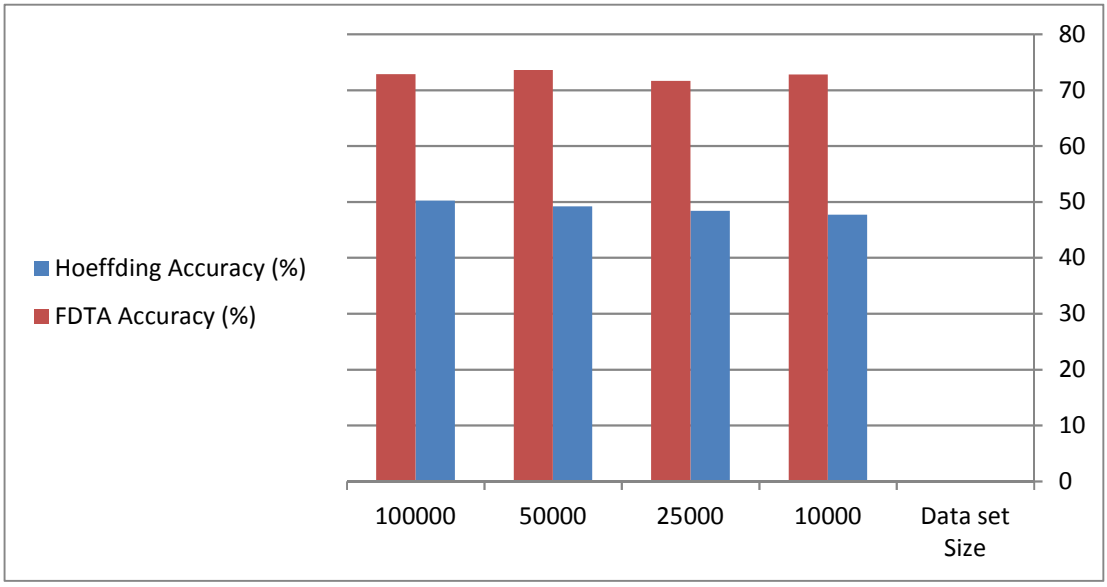

Figure 2 dataset-1 chart for Hoeffding vs. FDTA

From table (1.2) and figure 2 above one can see that FDTA has obtained higher accuracy than traditional Hoeffding tree algorithm.

Table 1.3 Hoeffding vs. FDTA for Data set-1

Data set size	Training records number	Testing records number	Hoeffding execution time (millisecond)	FDTA Execution time (millisecond)
10000	6667	3333	45	18
25000	16667	8333	82	33
50000	33333	16667	148	92
100000	66666	33334	194	165

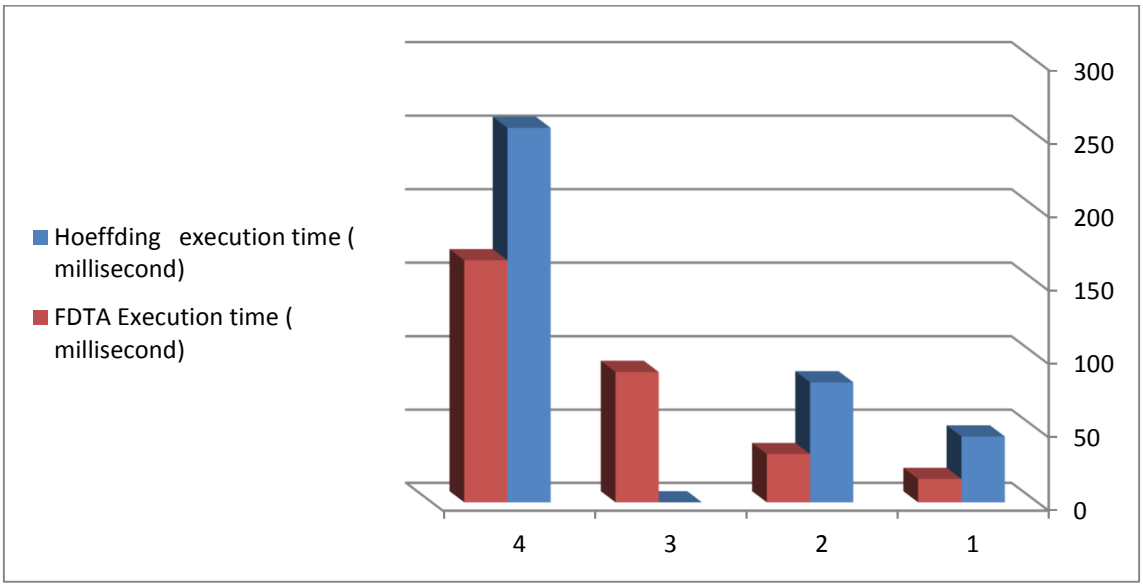

Figure 3 dataset-1 chart for Hoeffding vs. FDTA

As shown from table 1.3 and figure 3, FDTA has obtained lesser execution time comparing with the traditional Hoeffding tree algorithm.

Table 1.4 Hoeffding vs. FDTA for Data set-1

Data set size	Training records number	Testing records number	Hoeffding Memory space (byte)	FDTA Memory space (byte)
10000	6667	3333	11649	1990
25000	16667	8333	10953	3648
50000	33333	16667	27700	18800
100000	66667	33333	526680	550440

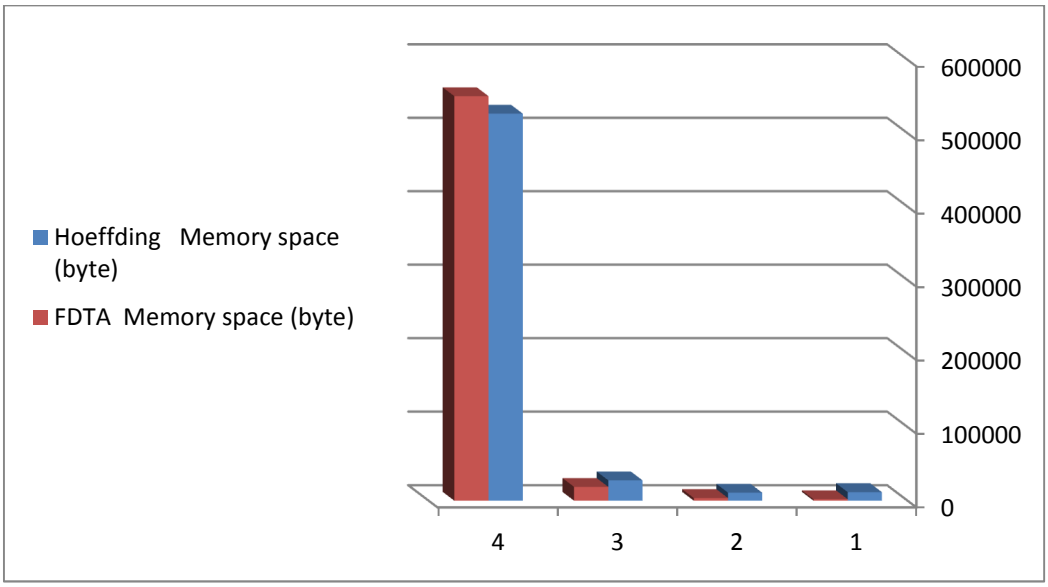

Figure 4 dataset-1 chart for Hoeffding vs. FDTA

As shown from table (1.4) and figure 4, FDTA has obtained lesser memory space comparing with the traditional Hoeffding tree algorithm in first three datasets sizes.

7 CONCLUSION

In this paper we have used two classification algorithms. The first algorithm is Hoeffding tree. Hoeffding trees a decision-tree learning method. Hoeffding trees can be learned in constant time per example while being nearly identical to the trees a conventional batch learner would produce, given enough examples. The name is derived from the Hoeffding bound that is used in the tree induction**.** The second one is a proposed algorithm is called fast decision tree (FDTA) which is an improvement classification method based on Hoeffding tree. The *t* is an algorithm parameter has shown an effect on a decision regarding making a split process on the tree or not. So *t* is generated in order fashion in domain (0-1) instead of treated as a fixed value as known in traditional Hoeffding tree (*t* known as 0.05). According to the obtained results above, it shows that FDTA is gained highest accuracy than Hoeffding tree, the same things regarding memory space and execution time.

REFERENCES

[1] Homayoun, S. and Ahma,dzadeh M. " A review on data stream classification approaches" Department of Computer Engineering and Information Technology, Shiraz University of Technology, Shiraz, Iran . *Journal of Advanced Computer Science & Technology, 5 (1) (2016) 8-13.*

[2] J. Gama, *Knowledgeb Discovery from Data Streams*: Chapman Hall/CRC, Taylor & Francis Group, 2010.

[3] J. G. M. M. Gaber, Learning from Data Streams: Springer, 2007.

[4] M. Kantardzic, Data mining: concepts, models, methods and algorithms: Wiley-IEEE Press, 2011.

[5] O. M. L. Rokach, Data Mining and Knowledge Discovery Handbook, 2 ed.: Springer, 2010.
[6] S. agrawl and B. R. parasad, "high speed streaming data analysis of web generated log stream " in 2015 IEEE 10th international conference on industrial and information systems (ICIIS).
[7] D. B. S. P. M. Hanady Abdulsalam, "Classification Using Streaming Random Forests," *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* vol. 23, p. 15, 2011 .
[8] Domingo's, P. and Hulten, G., (2000): *Mining High-Speed Data Streams*. In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining.
[9] G. hulten, l. spencer and p. Domingos," mining time- changing data stream" 2001. In proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining.
[10] Street, W. nick and yongseog Kim. "A streaming Ensemble Algorithm SEA for large scale classification". In proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, 2001.
[11] J. Gama , P. M edas and R. Rocha," forest trees for online data " , In SAC 04, In Proceeding of 2004 syposium on applying computing pages 632-636 new york, USA, 2004.
[12] Cohen, Lior, Gil Avrahami, Mark Last and Oscar Kipersztok, "Real Time Data for Non-Stationary Data Stream from Sensor Network", information fusion, vol, 9. 2008.
[13] Mena- Torres D., Jesus S. A. "Similarity- Based Approach for Stream Data classification" Expert System with Applications, 2014.
[14] "Review on Data Stream Classification Algorithm" International Journal of Conceptions on Electrical and Electronics Engineering Vol. 1, Issue 1, Oct 2013; ISSN: 2345 – 9603
[15] Babcock, B., Babu, S., Deter, M., Motwani, R., and Widom, J., (2002): *Models and issues in data stream systems*. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS). Madison, Wisconsin, pp. 1-16.
[16] C. W. Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen & Nikunj C. Oza, "Facing the reality of data stream classification: coping with scarcity of labeled data," *Knowl Inf Syst,* vol. 33, p. 32, 2011.