# An Efficient Indexing Structure for Ensemble Classification of Data Streams Using Forest-Tree Mechanism

Priyanka Narayan Kamble, Prof. Sonali A. Patil

[1] *Student, Computer Engineering, Bhivarabai Sawant Institute of Technology & Research Wagholi, Pune, Maharashtra, India*

[2] *Assistant Professor, Computer Engineering, Bhivarabai Sawant Institute of Technology & Research Wagholi, Pune, Maharashtra, India*

## ABSTRACT

*Ensemble learning is used for data stream classification, as it facing problem to large size of stream data and concept drifting. Direct output of an extensive number of base classifiers in the troupe amid expectation keeping group gaining from being viable for some true time critical data stream applications, e.g. Web traffic. In this data streams usually come at a speed of GBPS, and it is important to order every stream record in a timely manner. Thats why we propose a novel E-tree indexing structure to sort out all bases in an ensemble for fast prediction and using random forest classifier-trees regard groups as spatial databases and utilize an R-tree to less the expected prediction time from direct to sub-straight multifaceted nature. E-trees can be changed by continuously integrating new classifiers and discarding outdated ones, well adjusting to new patterns. Our system works on web traffic stream monitoring to classify web pages and extend work to classify traffic stream data.*

**Keyword: -** *Stream data mining, classification, ensemble learning, spatial indexing, and concept drifting*

## 1. INTRODUCTION

Information stream grouping speaks to a standout amongst the most imperative undertakings in information stream mining which has been prevalently utilized as a part of constant interruption discovery, spam sifting, and malicious site observing. In the applications, information arrive persistently in a stream design, auspicious forecasts in distinguishing malignant records are of key significance. Contrasted with customary order, data stream classification is confronting two additional difficulties: substantial/expanding information volumes and floating/developing ideas. To address these difficulties, numerous outfit based models have been proposed as of late, including weighted classifier outfits. Incremental classifier outfits classifier and bunch gatherings to give some examples. While these models shift starting with one then onto the next, they share striking closeness in their configuration: utilizing separate and-overcome systems to handle substantial volumes of stream information with idea floating. In particular, these models segment constant stream data into little information lumps, assemble one or various light-weight base classifier(s) from every lump, and join base classifiers in various courses for forecast. Such an outfit learning plan appreciates various points of interest, for example, scaling admirably, adjusting rapidly to new ideas, low fluctuation blunders, and simplicity of parallelization. Accordingly, group has gotten to be a standout amongst the most well-known strategies in information stream classification. To date, existing deals with group learning in information streams for the most part concentrate on building precise gathering models. Expectation productivity has not been concerned essentially since

(1) Expectation commonly takes straight time, which is adequate for general applications with undemanding expectation productivity.

(2) Existing works just consider consolidating a little number of base classifiers, e.g., close to 30.

In any case, there are increasingly more real world applications where stream data arrive seriously in expansive volumes. Also, the shrouded designs underneath information streams might change ceaselessly, which requires an expansive number of base classifiers to catch different examples furthermore, frame a quality group. Such applications call for quick sub-direct expectation classifications. Propelling case, in online site page stream checking, learning can be utilized to recognize malignant pages from ordinary pages, both arriving consistently, in ongoing.

## 2. RELATED WORK

Peng Zhang, Chuan Zhou, Peng Wang, Byron J. Gao, Xingquan Zhu, and Li Guo [1] suggest Ensemble learning is a common tool for data stream classification, mainly because of its inherent advantages of handling large volumes of stream data and concept drifting. Previous studies, to date, have been primarily focused on building accurate ensemble models from stream data. However, a linear scan of a large number of base classifiers in the ensemble during prediction incurs significant costs in response time, preventing ensemble learning from being practical for much real-world time critical data stream applications, such as Web traffic stream monitoring, spam detection, and intrusion detection.

P. Sravanthi , J. S. Ananda Kumar [2] introduce Ensemble stream data management technique ,is one of the stream data mining techniques. Ensemble-tree is an indexing data structure for storing classification rules of ensemble classifiers.

Jiong Zhang, Mohammad Zulkernine, and Anwar Haque suggest In[3] anomaly detection, novel intrusions are detected by the outlier detection mechanism of the random forests algorithm. After building the patterns of network services by the random forests algorithm, outliers related to the patterns are determined by the outlier detection algorithm. The hybrid detection system improves the detection performance by combining the advantages of the misuse and anomaly detection. We evaluate our approaches over the Knowledge.

Discovery and Data Mining 1999 (KDD'99) data set. ATA stream classification represents one of the most important tasks in data stream mining [4], [5], which has been popularly used in real-time intrusion detection, spam filtering, and malicious website monitoring. In the applications, data arrive continuously in a stream fashion, timely predictions in identifying malicious records are of essential importance.

## 3. SYSTEM ARCHITECTURE

This section narrates about the techniques that we are included in web traffic data streaming classification using E-tree techniques. And the whole process is depicted in the figure 1.

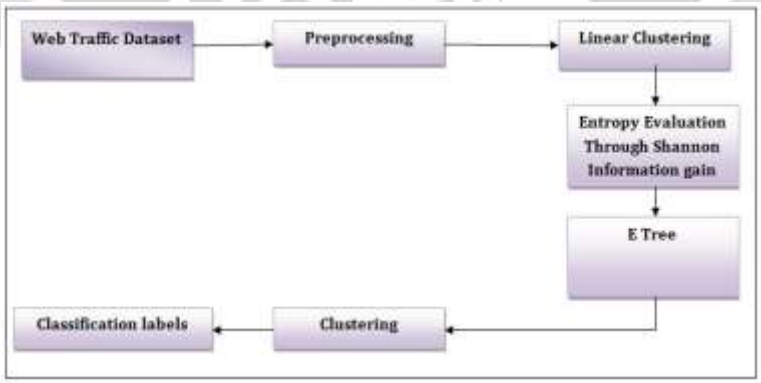


**Fig 1:** System Overview of web traffic data streaming classification system

The steps that are depicted in the system overview of figure 1 can be elaborated using the following steps.

### 3.1 Preprocessing

This is the initial step of the proposed methodology where system is given input of web traffic data log files that is collected from the URL http://recsys.yoochoose.net/challenge.html . Here this data set consists of some attributes like Session ID, Item ID, Product ID and date time. Here in this step of the model system reads the whole data set in the form of buffer string then it stores the each line of the buffer string in a list. Then this list is subjected to select some needed attributes like session id, Price id and item id. Once these attributes are selected then they are stored in a double dimensional list to process further.

### 3.2 Linear Clustering

Here in this step all the attributes that are stored in the two dimensional array is subjected to cluster linearly. By doing this distribution of the attributes can be verified using entropy evaluation process of the next step. This linear clustering can be depicted using the following algorithm of 1.

**ALGORITHM1: LINEAR CLUSTERING**

// Input : Set A ={ S,P,I }

// output : Set C={ $\{S_i\},\{p_j\},\{I_k\}$ }

Step 0: Start

Step 1: Get Set A

Step 2: Create a List T,Set Count=0

Step 3: For i=0 to size of A

Step 4: count++

Step 5:Add Ai to T

Step 6: IF Count=10

Step 7:Add T to C

Step 8:Set Count=0

Step 9:Empty T

Step 10:END FOR

Step 11: return C

Step 12: END

### 3.3 *Entropy Evaluation*

This is the process of evaluating the distribution factor of each attributes that eventually helps to create the high end Clusters that thereby helps us to get proper classification labels. This process begins with the process of identifying all the unique attributes and then to search these attributes for the presence of clusters. Then by using Shannon information gain theory entropy of the attributes are calculated using the equation 1.

$$IG(E) = -(X/T)\log(X/T) - (Y/T)\log(Y/T) \quad \text{--------(1)}$$

Where

*X= Number of the clusters where attribute is present*

*Y= Number of the clusters where attribute is not present*

*T=Total number of clusters*

*IG (E ) = Entropy of the given attribute using Information Gain theory .*

Here the Information gain value that eventually represents the distribution factor of the attributes. This entropy value is varied from 0 to 1. If any attribute is having entropy nearer to 1 means that is been distributed more in the web traffic data. And if this entropy value is nearer to 0 means the respective attribute is very least distributed over the web traffic data.

**3.4 E- Tree Formation and indexing**

This is the step where an improved Ensemble tree is created using the distribution factor of the attributes that are estimated in the prior step. Here the first attribute is fixed as the root node and all the attributes are fixed on their locations based on the scale of the entropy value. And the attributes which are having less entropy value that of the root node is assigned to the left child . If attributes are having higher entropy value that of root node then they are assigned to the right child. If the attributes are having the entropy as of any existed node in the created ensemble Tree then it is accumulated in the same cluster of the node based on the index of the tree nodes. And this process of creation of E- tree is depicted in algorithm 2.

**Algorithm 2:** E- tree

//input : Attribute Set A={ $A_t$,$E_n$}

Where

$A_t$ is Attribute value

$E_n$ is Entropy value

// output : E tree $E_T$

Step 0: Start

Step 1: Create an empty tree as **T**

Step 2 :**FOR** i=0 to size of **A**

Step 3: IF i==0

Step 4: Create the Root Node for first Attribute $\mathbf{R_n}$

Step 5: END IF

Step 6: ELSE

Step 7: get $A_i$ and $E_i$

Step 8: Compare $E_i$ with the instance root node $R_n$

Step 9:**IF** ($\mathbf{E_i}$ support < $\mathbf{R_n}$)

Step 10:**IF** $E_i$ Є **T**

Step 11:Then add to the $A_i$ index node

Step 12: **ELSE**

Step 13: Add node as left child in **T**

Step 14: **Else**

Step 15:**IF** $E_i$ Є T

Step 16:Then add to the $A_i$ index node

Step 17: **ELSE**

Step 18: Add node as right child in **T**

Step 19:End **FOR**

Step 20: return **T**

Step 21: Stop

**3.5 Clustering and Classification Label formation**

This is the step where created E Tree is traversed in pre ordered manner to fetch all the similar node entities which in turn gives rise to list of elements. Then these elements are been in stored separate list to create the clusters of web traffic data along with the entropy values. Finally these entropy values are gathered and the respective attribute names are collected with their unique inherited values to get the classification labels through generated E-tree.

## 4. RESULTS AND DISCUSSIONS

### 4.1 Experimental Setup

Proposed system of E-tree for web traffic data streaming is deployed using java based windows machine. System hails the configuration of Core i3 Pentium processor with 2 GB primary memory. System uses Netbeans as standard Development IDE. Proposed model uses the web traffic data set collected over the URL http://recsys.yoochoose.net/challenge.html. This contains 5 attributes and thousands of instances.

### 4.2 Performance Evaluation

Proposed model of generation of Web traffic data streaming classification label is evaluated based on the two well-known and most effective parameters in data mining like precision and recall. Relative preciseness of the system is calculated from the precision parameter and the relative relevance of the system is calculated using recall of the system. To know these parameters in depth they can be elaborated as follows

Let ..

$R_L$ – Relevant Classification labels are generated for the given attribute.

$IR_L$ – Irrelevant Classification labels are generated for the given attribute

$R^I$ – Relevant Classification labels are not generated for the given attribute

Then Precision can be stated as shown in equation 2

$$P_r(x) = ( R_L / (R_L + IR_L) ) / 100 \text{ ----------(2)}$$

Where Pr(x) – Precision Function

So, Recall can be stated as shown in equation 3

$$R_c(x) = ( R_L / (R_L + R^I ) ) / 100 \text{ ----------(3)}$$

Where $R_c(x)$ – Recall Function

When an experiment is conducted based on this to measure F1 , which gives more intrinsic values and it can be measured using equation 4

$F1 = ( ( P_r \times R_c) / ( P_r + R_c)) \times 2$ ----------(4)

F1= F measure

$P_r$= Precision

$R_c$= Recall

On conducting experiment for different number of the classified labels system found some interesting facts of precision and Recall as tabulated in table1.

| No of Actual Existed Classification Labels | R | IR | R' | Precison | Recall | F1 |
|---|---|---|---|---|---|---|
| 5 | 5 | 0 | 0 | 100 | 100 | 100 |
| 10 | 8 | 1 | 1 | 88.88889 | 88.88889 | 88.88888889 |
| 15 | 13 | 1 | 1 | 92.85714 | 92.85714 | 92.85714286 |
| 20 | 16 | 2 | 2 | 88.88889 | 88.88889 | 88.88888889 |

**Table 1**: Precision, Recall and F measure Evaluation

When F measure of our system is compared with that of [x] , Which is designed for classification label identification for novelty data some facts are revealed as tabulated in table 2.

| Methodology | F1 |
|---|---|
| OC -SVM | 0.7705 |
| SVDD | 0.7907 |
| Entropy based E Tree | 0.926 |

**Table 2**: F1 Score Comparison

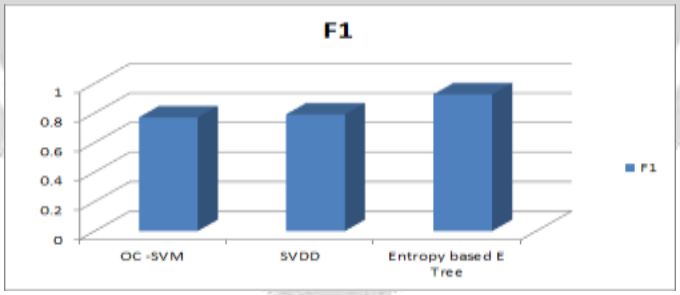


**Fig 2**: Performance Comparision of F1 Score

On observing the plot in figure 2 it is clearly indicating that our proposed system of classification based on the entropy for E-tree creation yields more than that of mentioned two methodologies.

## 5. ACKNOWLEDGEMENT

The authors would like to thank the researchers as well as publishers for making their resources available. We are very much thankful to our Principal Dr.T.K.Nagraj & HOD Prof. G.M.Bhandari. We are also thankful to our PG Coordinator Prof. A.C.Lomate& Project Guide Prof. Sonali A. Patil. We are thankful to the authorities of Savitribai Phule University of Pune, for their constant guidelines and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

## 6. REFERENCES

[1]Peng Zhang, Chuan Zhou, Peng Wang, Byron J. Gao, Xingquan Zhu, and Li Guo"E-Tree: An Efficient Indexing Structurefor Ensemble Models on Data Streams" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 2, FEBRUARY 2015

[2]P. Sravanthi and J. S. Ananda Kumar"Extended R-Tree Indexing Structure for Ensemble Stream Data Classification "International Research Journal of Engineering and Technology (IRJET).

[3]Jiong Zhang, Mohammad Zulkernine, and Anwar Haque "Random-Forests-Based Network IntrusionDetection Systems" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 38, NO. 5, SEPTEMBER 2008

[4] P. Zhang, J. Li, P. Wang, B. Gao, X. Zhu, and L. Guo, "EnablingFast Prediction for Ensemble Models on Data Streams," Proc. 17thACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining(KDD), 2011.

[5] C. Aggarwal, Data Streams: Models and Algorithms. Springer, 2006.

[6] Matthew Martinez , Phillip L. De Leon and David Keeley ," NOVELTY DETECTION FOR PREDICTING FALLS RISK USING SMARTPHONE GAIT DATA" ,IEEE ,978-1-5090-4117-6 - 2017.