RESEARCH ARTICLE OPEN ACCESS

"An approach to evaluate Data Stream Mining with Decision Tree induction method in Machine Learning"

Vijay Shukla*, K P Yadav**
**(Department of Computer Science & Engineering, ITS Engineering College, Greater Noida, India*
*** (Department of Computer Science, Sangam University, NH-79, Bhilwara chittor Bypass, Bhilwara, Rajasthan, India*

ABSTRACT
One way to deal with induction is to build up a decision tree from a bunch of models. At the point when utilized with loud as opposed to deterministic information, the strategy includes three primary stages - making a complete tree, ready to group all the models, pruning this tree to give factual dependability. This paper is worried about the primary stage - tree creation - which depends on a measure for "integrity of split," that is, the way well the attributes segregate between classes. The results show that the decision of measure affects the size of a tree yet not its accuracy, which remains the same in any event, when attributes are chosen randomly. Data Stream mining is professed to be the successor of conventional data mining where it is equipped for mining constant approaching data streams in real-time with a satisfactory performance. Many PC applications developed to on the web and on-request premise, new data are taking care of in at high rates. DSM then again dynamically fabricates and reestablishes the decision tree model when another cruise of information stop by. In this paper, we assessed the presentation of a mainstream decision tree is called as Hoeffding Tree opposite of C4.5. A decent blend of sorts of datasets was utilized in the tests for exploring the obvious contrasts between the decision trees. An open-source DSM test system was customized in JAVA for the trials.

Keywords- acquisition data stream mining, decision trees, Hoeffding tree algorithm, JAVA simulator, noise data

I. INTRODUCTION

There are various ways to deal with inductive learning (Michalski, Carbonell, and Mitchell, 1983, 1986; Bratko and Lavrac, 1987), one of which includes the construction of decision trees. This strategy was developed initially by Chase, Marin, and Stone (1966) and later adjusted by Quinlan (1979, 1983), who applied his ID3 algorithm to deterministic spaces, for example, chess end games. Breiman, Friedman, Olshen, and Stone (1984) independently developed a comparable way to deal with grouping. Data stream mining may address the difficulties of preparing high-volume, constant data with cautious setups of the mining boundaries. Helpful examples can be extricated from the most refreshed data; anyway they are not consistent but rather truly changing with the expanding data input. In the event that the data arrives in a fast, the essential extension to get ongoing knowledge is the given data rate should be not exactly the pace of mining measure. Despite the fact that the accuracy of DSM relies upon the given resources[1], for example PC's capacity, it is additionally identifying with the framework architecture and algorithm granularity.

DSM might be actualized across a wide scope of applications where information stream in quickly and the all out information size might be endlessly huge. Applications remember financial analysis for stock market, network interruption identification, web personalization, online snap streams analysis, etc as referenced in. Nonetheless, among the different stream mining algorithms, there are not many investigation focusing on how stream mining's exhibition when they are applied for real-time application. Along these lines, we construct a stream mining simulation system in JAVA open source design[2]. In this paper, our exploration centers around decision tree calculation in stream mining since decision tree is perhaps the main classification techniques and it has a more serious level of interpretability than others.

So far the best of the creators' information no inside and out investigation has been done on evaluating the performance of HTA in correlation with customary decision trees, particularly by utilizing enormous estimated datasets. Specifically, we analyze the impacts of commotion in information in relation to the decision trees. Three distinct sorts of datasets of various natures and sizes were utilized in the experiments. The remainder of the paper is organized as follows: Section 2 gives a foundation presentation of decision tree algorithm in stream mining, and records a few difficulties that the current calculations are confronting. Section 3 is

about the plan of our simulation system, and the major practical parts. Section 4 shows the three trials directed under the simulator, trailed by discussions of the results.

II. FOUNDATION OF DECISION TREE ALGORITHM

It very well might be adequate to utilize only a little accessible information test for picking the split quality at some random hub for a decision tree. This statistical method is known as Hoeffding bound or additive Chernoff bound, which is utilized to take care of the troublesome issue of choosing precisely the number of tests are important at every hub by utilizing a statistical result [7,8,9,10,11,12]. VFDT (Very Fast Decision Tree) system [5] develops a decision tree by utilizing steady memory and consistent time per test. It is a pioneer prescient procedure that utilities Hoeffding bound. The tree is worked by recursively supplanting leaves with decision nodes. The adequate statistics of attribute esteems are put away in each leaf. Heuristic evaluation function is utilized to decide split attributes changing over from leaves to nodes. Nodes contain the split attributes and leaves contain just the class labels. The leaf speaks to a class that the sample labels. At the point when an example enters, it crosses the tree from root to a leaf, assessing the pertinent attribute at each and every node. After the example arrives at a leaf, the adequate statistics are refreshed. As of now, the framework evaluates every conceivable condition dependent on attribute values, if the statistics are sufficient to help the one test over the others; a leaf is changed over to a decision node. The decision node contains the quantity of potential values for the picked attribute about the split-test introduced.

The primary components of VFDT include:

Initially, express the tree just has a single leaf - the root of the tree. Furthermore, characterize the heuristic assessment work (denoted by G(.)), which assembles a decision tree with Information Gain like ID3 [6]. The Information Gain measures that amount of information which is necessary to classify a sample that reaches the node in terms of Equation.

The sufficient statistics estimates the merit of a discrete attribute's counts nijk, representing the number of samples of class k that reach the leaf where the attribute j takes the value i. The information of the attribute j is given by Equation 2, where Pik is the probability of observing the value of the attribute i given class k. Pi in Equation 3 is the probabilities of observing the value of attribute i.

$$G(A_j) = \text{info(samples)} - \text{info}(A_i) \quad (1)$$

$$\text{info}(A_j) = \sum P_{ij} \left(\sum - P_{ik} \log(P_{ik})\right) \quad (2)$$

$$P_{ik} = n_{ik} / \sum n_{jk} \quad (3)$$

$$P = \sum n / \sum \sum najb \quad (4)$$

$$\varepsilon = (R^2(\ln(1/\delta))/2N)^{1/2} \quad (5)$$

For 'n' number of real-valued random variable 'r' the rang of random number is defined as R. Hence, Hoeffding bound can be calculated as mentioned in Equation 5. It illustrates that with confidence level $(1 - \delta)$, the true mean of *r* is at least r, which is known as observed mean of samples. For a probability the range *R* is 1, and for an information gain the range *R* is $\log_2 Class\ \#$.

An important part of VFDT is the use of Hoeffding bound to choose a split attribute as the decision node. Let x_a be the attribute with the highest G(.), x_b be the attribute with second-highest G(.).

Therefore $\otimes G = G(x_a) - G(x_b)$ is the difference between the two top quality attributes. If $\otimes G > \varepsilon$ with N samples observed in leaf, while the Hoeffding bound states with probability $1 - \delta$ that x_a is the attribute with highest value in G(.). Then the leaf is converted into a decision node which splits on x_a

III. WEB-BASED STREAM MINING SIMULATION

The simulation is a JAVA-put together framework based with respect to J2EE design. The UI is a bunch of web pages that can be effectively gotten to through web browsers. The framework is built by four significant utilitarian components: Data Source, Trigger, Miner and

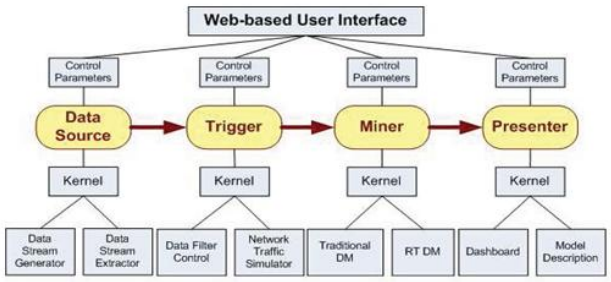

Figure 1.Web based JAVA stream mining simulation system

Presenter as shown in Figure 1 provides an online UI to setup the control parameters.

1st component: Data Source

This unit is utilized to get ready data contribution for mining. It has two kernels – data

stream generator and data stream extractor. Generator is utilized to create manufactured data and extractor is utilized to gather genuine world preprocessed data maybe that ought to be tweaked in organizations. Information generator can recreate artificial datasets such LED, SEA, Wave, etc that come in different types of patterns. It likewise requires setting of some control parameters, for instance, commotion rate, missing information rate, property number, class number, etc. Information extractor is mindful to import input information from outer sources, for example, SQL database, excel, CSV file, etc.

2nd component: *Trigger*

This component can filter the missing data as instructed by the user's configuration. Generally, user can select whichever appropriate method on the best way to manage missing information: probabilistic split, total case method, amazing mode/mean imputation, separate class, surrogate split, and complete variation. Not with standing missing data filter; trigger component is additionally used to reproduce the network traffic adding to the gathered data source. The performance of genuine online data mining applications is known to be delicate to fluctuating network traffic. Bursty model is a fundamental information model for creating Internet traffic transfers. The simulator can produce such traffic of irregularity in information source generator. Client can characterize two parameters – p and L – to control Bursty traffic. p is the normal load of one bursty source which is characterized as the portion of time allotments this source spends in the dynamic state. L is the mean length of burst information stream. This length can be perceived as a normal unit packet size of TCP/IP protocol

3rd Component: Miner

This part is the center of recreation framework. It is an open platform on which more algorithms can be added. In this experiment, it contains decision tree algorithm, HTA, which speaks to stream mining. Traditional decision tree algorithm like ID3, C4.5 that depend on data pick up [2] are introduced in the simulator also. Other stream mining algorithms will before long be added to the recreation

4th Component: Presenter

After the mining process, simulation system presents the outcomes through this part. The outcomes can be introduced in various formats; some mainstream ones are charts and figures. To encourage a superior interpretable 'generally' picture to client, dashboard techniques are additionally applied in this system.

IV. RESULT ANALYSIS ON SIMULATION

The configuration of simulation system is 2.99 GHz CPU and 1 GB RAM along with the running environment JAVA SDK 1.6 with Apache Tomcat.

In this paper, we plan to run the reenactment framework more than three investigations. The first uses manufactured medium size dataset adulterated with different control levels of commotion. This trial is to examine the contrasts between conventional C4.5 decision tree and HTA stream mining when all is said in done situations where commotion irritation is a piece of messy information. The subsequent examination utilizes true little measured datasets. Investigation result is like that of the previous test. The third examination utilizes generally enormous measured engineered dataset and it shows mostly that customary decision tree calculation can't adapt. Through utilizing both ostensible and numeric information.

types, we examine the HTA performance regarding various sorts of dataset.

Some performance terms utilized in the experiments are explained here. Precision is the level of effectively grouped occurrences over the entire populace of cases. Tree size is the quantity of hubs in a tree model prior to pruning. Computation time is the net time taken for the tree model to be built in the simulation framework during the training stage, barring preprocessing and model validation.

4.1 Synthetic Medium Dataset Analysis

The information utilized in the analysis is artificially created by our simulation framework, by the Data Source segment. The reproduced model is three LED datasets, which contain 0%, 10 % and 20% measure of irregular clamor in the information individually. Inside an example there are seven ostensible credits with double qualities, and a class mark of various qualities. The investigation results show the perspectives on tree size and computation time.

A decision tree is assembled steadily as the data are divided by figuring out the highlights of an occasion the tree from the root to leaf hubs. A compact tree is wanted as frequently a more modest tree yields succinct standards. Figure 2 shows C4.5 tree sizes in various extents of noise in the datasets. When the dataset is sans noise, the tree size is a consistent. This number is tiny when contrasted with the tree size under noise-instilled data. It is we realized that C4.5 is touchy to noise. The presence of noise in the classes influences impressively the presentation of a classifier, since it grows the class boundaries that makes it turns out to be more hard to

decide them. The noise perturbation in the occurrences causes the classifier doles out wrong classes to examples that are accurately named, and the other way around. In the experiment we can see that boisterous occurrence directly affect the tree size and accuracy. The issue floats by the development of the cases in which the measure of instilled noise increments proportionally.

Similarly, HTA tree size is additionally developing as the noise level scales up as appeared in Figure 3. Clearly, the consequence of the principal experiment shows that the tree size of HTA is a lot more modest in hoards than that of C4.5 development. That suggests HTA is heartier than C4.5 for the hubs that are needed to stay in stack memory during the tree development. Another intriguing marvel that we found about HTA is that, when the noise level transcends half to 100%, the bend scopes as far as possible however flips down backward heading. At the point when the noise moves toward practically 100%, the exhibition bit by bit turns into that of the without noise data. This shows that the calculation basically gets befuddled in recognizing which are noises and genuine data. This can be clarified that characteristically HTA depends on Chernoff bound that has the numerical property of assessing the quantity of tosses that are needed to decide a head or tail (undifferentiated from flipping a coin) with a necessary assurance.

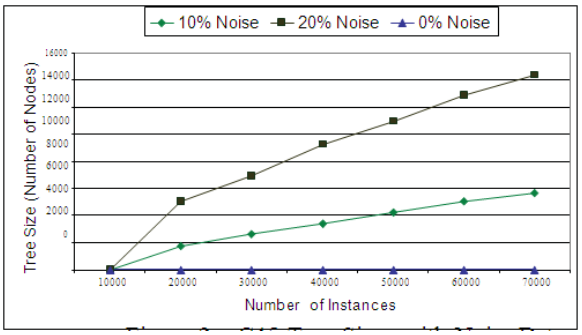

Figure 2. C45 Tree Sizes with Noise Data

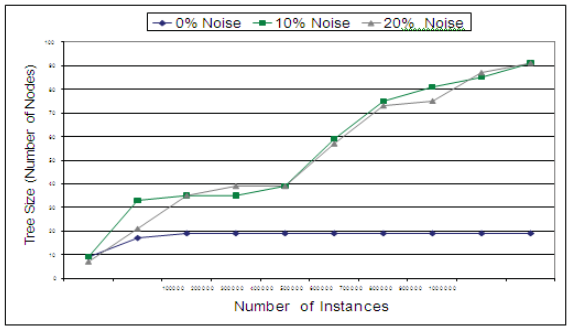

Figure 3. HTA Tree Size with Noise Data

The computation time is another indicator for data mining algorithm evaluation that is important to applications which require very less amount of data analysis latency.

It is well known that computation time is related to the data size. When more data has to be processed then longer time is required for its processing. An apparent situation is that C4.5 consumes more time than HTA when there is an increase in data size as shown in figure 4. When computation time is taken, say 6 seconds for processing on the same data set with noise of 20%, then C4.5 deals for 50,000 instances only same is mentioned in figure 4, and in the comparison of this while HTA has processed approximately 350,000 as indicted in figure 5. Though both algorithms are having same nature i.e. linear increase in computation time, C4.5 still provides much more computation time with respect to other.

4.2 E-Bay Auction Dataset Analysis

Other than engineered data, our simulation framework can imports certifiable data The data depicts the costs, time, and the auction introduction data on e-Bay of three kinds of commercial MP3 products (IPod, IPod small, IPod Nano). The objective class name is to foresee whether the accomplished deals income is more noteworthy than the normal deals income of the product classification. This speaks to a moderately complex data that are established from information about the vender, utilized in posting the auction, the auction strategy.

The result analysis is such that the computation time is related to size of data. When data size is increased then processing time is also get increased. An apparent situation that HTA while the data size is growing. Comparing with the same computation time, for example, 7 seconds for processing the same set of data with 30% noise, C4.5 can deal with only 13,000 instances as mentioned in figure 7, while HTA can process as many as 450,000 as mentioned in figure 8. Although both algorithms tend to take a linear increase in computation time, the rate of increase for C4.5 is much greater.

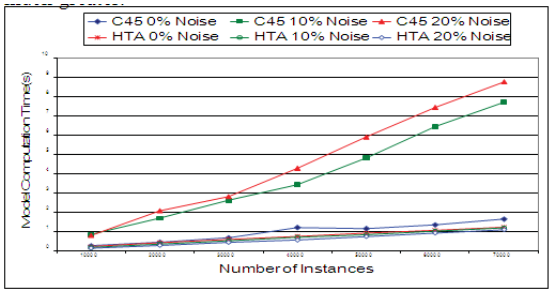

Figure 4. C45 and HTA Computation Time

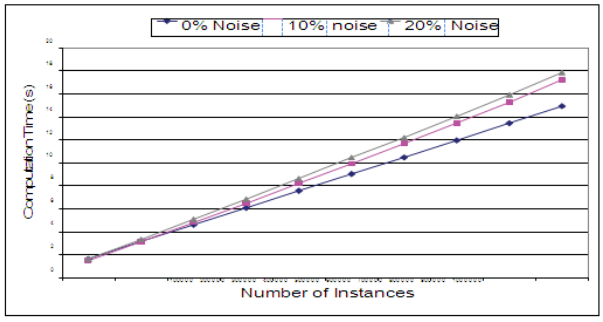

Figure 5. HTA Time of Large Data Size

On applying C4.5 and HTA, the outputs results the connection between accuracy versus sizes of tree size and instance numbers of IPod items as mentioned in figure 6 and figure 7. Comparative examples are shown across the all items. The examples overall show that the accuracy of C4.5 is higher than HTA because of multi-scanning across dataset of small size. The quantity of hubs in C4.5 tree is still more than HTA tree which shows that C4.5 can result in great accuracy that could have the asset to for the full HTA tree in the arrangement of data.

The result shown in example illustrates that C4.5 can achieve good accuracy it could have the resource to crunch over the full set of data. However, when we answer the question that between tree size and noise which factor HTA more significantly then we obtain the result which is more interesting and shown in figure 7 that as of increasing the number of instances the size of nodes are increasing in a significant manner which can be easily managed data through algorithmic approach

Table 1. E-Bay Auction Datasets

Name	Numeric Attr#	Nominal Attr#	Class#	Instance#
IPod	8	14	2	2,886
IPod Mini	8	14	2	2,965
IPod Nano	8	14	2	2,149

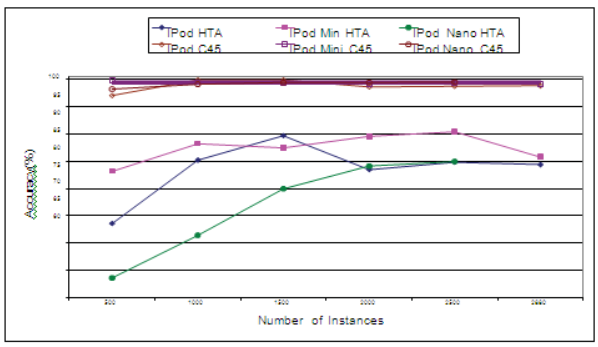

Figure 6. E-bay Data: C45 and HTA Algorithm Accuracy

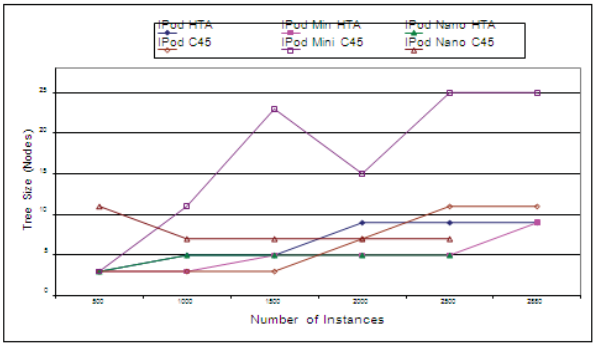

Figure 7. E-bay Data: C45 and HTA Algorithm Tree Size

4.3 Very Large Synthetic Dataset Analysis

C4.5 gets a preferred performance over HTA in small and medium sized dataset, for large sized C4.5 neglects to adapt to large data size due to the idea of the enlistment calculation requires tree hubs to be put away in stack memory has an actual breaking point, yet HTA can do with gigantic datasets, hypothetically with size way to deal with endlessness. In this test, we increase the experimental data size to more than 200,000,00, which is obtained by the framework of simulation. These parameters are mentioned in Table 2.

Table 2. Synthetic Large Datasets

Name	No of Attr	Attr Value	Noise percentage	No of Class	No of Instances
LED 7	7	Nominal	0 ~ 50%	10	10,000,000
LED 24	24	Nominal	0 ~ 50%	10	10,000,000
SEA	3	Numeric	0 ~ 50%	2	10,000,000

The time computation for a large datasets versus noise levels has been shown in figure 8. In general, both nominal and numeric datasets have a similar increasing computation time. While more percentage of noise never correlated for longer computation time. Small data computation time results more sensitive to noise as compared to numeric data. But computation time in both cases does not has any significant effect.

HTA accuracy of large datasets versus noise levels has been shown in figure 9. Generally, higher valued noise always results in low accuracy. When we compare both nominal as well as numeric datasets, then nominal data set has increased whereas accuracy. In Internet-based application without concept-drift stream mining, the noise data may be an important factor that causes accuracy decreases. The tree size of large nominal and numeric datasets has been shown in figure 10. The amount of noise percentage also increases the tree size of a node. When we compared the nominal data tree size versus numeric data, we found that nominal data tree

size is more sensitive to noise data than numeric data as shown by simulation result.

V. CONCLUSION

Hence we have seen that Stream mining is a complex data mining strategies, for its acclaimed preferences of taking care of data streams in a single pass at rapid. We have used JAVA open source architecture for the simulation. The simulation incorporates a data source generator and extractor, an organization traffic test system, a data excavator, just thus moderator. The plan is adaptable for new algorithms can simply connected and clients can work through web browser.

This article, announces the exploratory examination of C4.5 and HTA. The results are finished up as follow: (1) Used medium manufactured dataset to reproduce a new tree algorithm. The outcome shows C4.5 can accomplish a higher exactness than HTA. In any case, HTA works in quicker calculation time and more modest tree size than C4.5. (2) Used certifiable little dataset to look at C4.5 and HTA. The outcome is like the medium engineered set of data. (3) C4.5 uncovers its cutoff points while dealing with large set of data. Both ostensible and numeric manufactured datasets of tremendous sizes are utilized with HTA. Simulation result discovers HTA exactness is touchy to commotion data. Tree size is expanding straightly when more examples show up. Numeric dataset brings about a more perplexing tree model yet it is more steady exactness than ostensible dataset in HTA.

From the results obtained it has been realized that because of human errors in organization there may be missing communication in terms of information and noise. We have reported from our experiments that decision tree can arrive at good accuracy in perfect test. Information, characterization and exactnesses for large set of data result relatively more influenced by an expanding presence of noise in the two instances of C4.5 and HTA. C4.5 beats HTA in little to medium datasets on the grounds that various examining of the full dataset is accessible. The problem is more serious in nominal data type than numeric. Adapting new techniques like neural networks and other machine learning techniques will be our future scope of research work to handle with noise and missing data.

REFERENCES

[1]. Domingos, P. and Hulten, G. Mining high-speed data streams. In Proceedings of the Sixth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, ACM, New York, 2000, pp. 71-80.

[2]. Hulten, G., Spencer, L., and Domingos, P. Mining time-changing data streams. In Proceedings of the Seventh ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, ACM, New York, 2001, pp. 97-106.

[3]. Gama, J., Medas, P., and Rodrigues, P. Learning decision trees from dynamic data streams. In Proceedings of the 2005 ACM Symposium on Applied Computing, ACM, New York, 2005, pp.573-577.

[4]. Tao Wang, Zhoujun Li, Xiaohua Hu, Yuejin Yan, and Huowang Chen. A New Decision Tree Classification Method for Mining High- Speed Data Streams Based on Threaded Binary Search Trees. Emerging Technologies in Knowledge Discovery and Data Mining. Springer. 2009, pp. 256-267.

[5]. Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby. New Options for Hoeffding Trees, Advances in Artificial Intelligence, Springer, 2007, pp. 90-99.

[6]. Nishimura, S., Terabe, M., Hashimoto, K., and Mihara, K. Learning Higher Accuracy Decision Trees from Concept Drifting Data Streams. In Proceedings of the 21st international Conference on industrial, Engineering and Other Applications of Applied intelligent Systems: vol. 5027. Springer-Verlag, Heidelberg, 2008, pp.179-188.

[7]. Gaber, M.M., Data Stream Mining Using Granularity-Based Approach, Studies in Computational Intelligence, Vol 206, 2009 pp. 47-66

[8]. Ding, Y. and Simonoff, J. S.. An Investigation of Missing Data Methods for Classification Trees Applied to Binary Response Data. J. Mach. Learn. Res. 11, 2010, pp. 131-170.

[9]. Hang Y. and Simon Fong, Investigating the Impact of Bursty Traffic on Hoeffding Tree Algorithm in Stream Mining over Internet, In proceeding of 2nd International Conference on Evelving Internet (INTERNET), 2010, Valencia, Spain.2010, pp.144–155

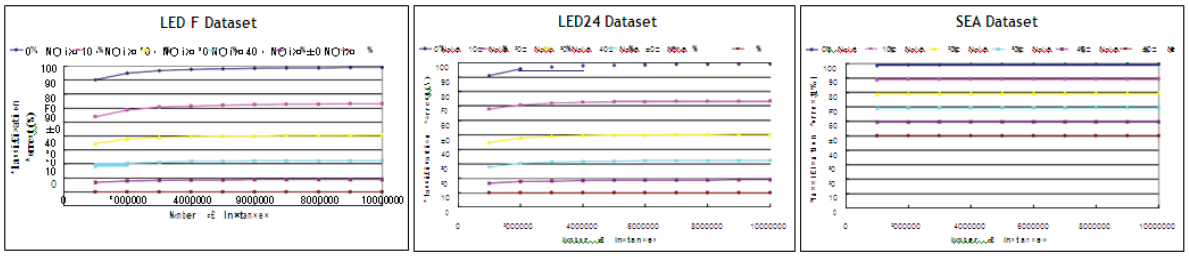

Figure 9. HTA Accurancy in Large Dataset

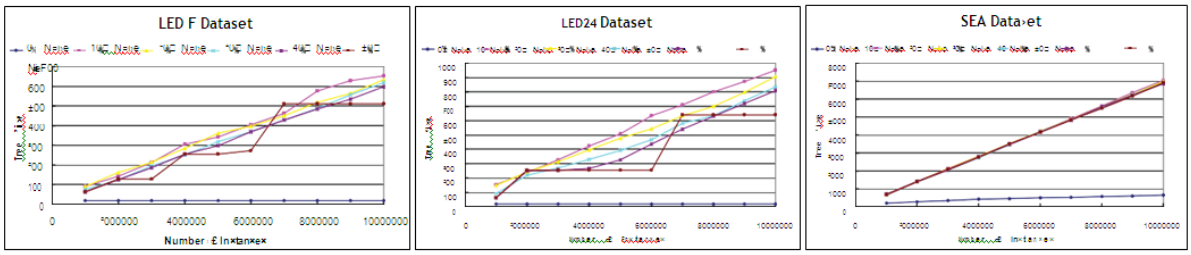

Figure 10. HTA Tree Size in Large Dataset

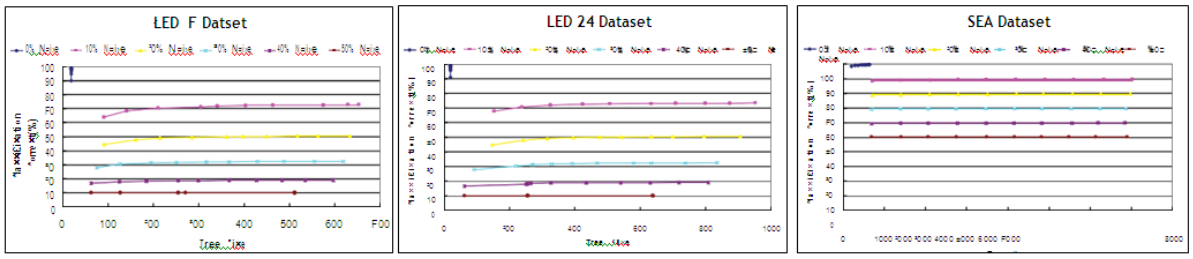

Figure 11. HTA Tree Size and Accuracy in Large Dataset