

# Enhancement of Very Fast Decision Tree for Data Stream Mining

Mai LEFA<sup>1\*</sup>, Hatem ABD-ELKADER<sup>2</sup>, Rashed SALEM<sup>2</sup>

<sup>1</sup> Department of Information System, Sadat Academy for Management Science, Cairo, Egypt  
mailefa12345@gmail.com (\*Corresponding author)

<sup>2</sup> Department of Information Systems, Faculty of Computers and Information – Menoufia University, Shebin Elkom, Egypt  
hatem6803@yahoo.com, rsalem@ci.menofia.edu.eg

**Abstract:** Traditional machine learning (ML) algorithms use static datasets to model knowledge. Nowadays, there is an increasing demand for machine learning based solutions that can handle very huge amounts of data in the shape of streams that never stop. The Very Fast Decision Tree (VFDT) is one of the most widely utilized data stream mining algorithms (DSM), despite the fact that it wastes a huge amount of energy on trivial calculations. The machine learning community has come first in terms of accuracy and execution time while designing algorithms like this. When assessing data mining algorithms, numerous types of studies include energy usage as a crucial factor. The purpose of this research is to create a hyper model to optimize the VFDT algorithm, which reduces the waste of energy while maintaining accuracy. In the proposed method, some fixed algorithm parameters were changed to dynamic parameters after analyzing each of them separately and knowing the extent of their positive impact on reducing energy consumption in several cases in algorithm. The practical experiment was conducted on both the algorithm in its basic form and the algorithm in the proposed form on several different types of datasets in the same application environment. The main advantage of the results of the proposed method compared to the results of the basic algorithm is that there was a noticeable development in the performance of the algorithm in terms of reducing its energy consumption and maintaining its accuracy levels.

**Keywords:** Data stream mining, Very fast decision tree algorithm, Hoeffding bound, Energy consumption, Massive online analysis.

## 1. Introduction

Recently, machine learning (ML) has changed the focus and application of its techniques. More and more attention shifts from off-line mining of static data towards ubiquitous mining and real-time applications dealing with data represented by streams. Data streams refer to a procedure in which examples occur continuously and possibly indefinitely through time (Kosina & Gama, 2013; Kok et al., 2019). Data stream mining is a chunk of machine learning that entails analyzing and building a prediction or classification algorithm on the basis of data flows that are continually growing as time series. In terms of objective, data stream mining is similar to traditional data mining but it is extremely different in terms of processing or executing the mining work (Fong et al., 2018; Krawczyk et al., 2017). The underlying challenges of infinite high-speed data streams are the cause of this difference. It renders traditional data mining algorithms and techniques incapable of properly processing data streams, needing the development of streaming data mining algorithms (Prasad & Agarwal, 2016). Various data stream mining tasks exist such as data stream classification, data stream clustering, and so on. This study will focus on multi-label data stream classification because it is the top highly used kind of data mining and a popular research area.

Classification is a method of classifying unknown data that usually involves two steps: training a generic model with known attributes and associated labels from a training data set, and then using the taught model to expect the labels of new data instances. Several data stream mining researches have been published lately introducing the specifics of a variety of data stream algorithms. The Very Fast Decision Tree (VFDT) is one of the supervised learning algorithms that can be applied to both classification and regression problems (Ibrahim & Abdulaziz, 2020), being the most commonly used algorithm in DSM Classification that extends the Hoeffding tree algorithm. It works well with convection data, and it is much better than traditional classification methods in terms of speed and accuracy but it wastes a huge amount of energy on trivial calculations (Zheng et al., 2019).

This work proposes a hyper model to reduce the overuse of algorithm's energy. The parameters of the algorithm should be improved for the energy saving. When some parameters are fixed, the algorithm calculates unnecessary operations. Therefore, the algorithm wastes energy on making such operations. So, the paper proposes dynamic parameters instead of the fixed ones to enhance Hoeffding tree parameter adaption and reduce the overuse of energy. Energy (joules) is the product

of power (watts) and time (seconds) as shown in Equation 1 (García-Martín, 2017).

$$Energy = Power \times Time \quad (1)$$

The paper is organized as follows. The VFDT algorithm and relevant researches are discussed in Section 2. Section 3 presents the theoretical hyper model of enhanced VFDT (E-VFDT). Section 4 presents experiments and results. Finally, Section 5 provides the conclusions and prospects for a future work.

## 2. Background and Related Work

### 2.1 VFDT

The algorithm under discussion is the VFDT. It is an online decision tree algorithm that can assess data from a flow, upgrade the model when fresh examples come, and read data only once (García-Martín, 2017). The instances are read one by one by the algorithm which sorts the proper leaves and changes the statistics for those leaves. To keep the statistics updated, the algorithm keeps a table with the observed attribute of each node. Each leaf saves the examples that have been observed so far (García-Martín et al., 2018).

The algorithm is used to determine the information gain ( $G$ ) for all the observed features after reading the minimum number of instances ( $n$  min) at that leaf. Let  $G(X_i)$  be the heuristic measure of attribute  $X_i$ .  $X_1$  is the attribute with the highest  $G(\cdot)$ , and  $X_2$  is the attribute with the second-highest  $G(\cdot)$ .  $\Delta G$  is the evaluation function which can be information gain, Gini index or gain ratio  $\Delta G = G(X_1) - G(X_2)$  (Domingos & Hulten, 2000).

The difference in information gain between the best and the second-best feature is compared to the Hoeffding Bound ( $\varepsilon$ ) after calculating it in Equation 2. If  $\Delta G > \varepsilon$ , a node will replace that leaf, and the best feature will be split. That feature is abolished from the list of features which can be used to split that branch (García-Martín et al., 2018). If  $\Delta G < \varepsilon$ , it signifies that the highest and second-highest  $G(\cdot)$  are not significantly different, and the process is stopped in a tie condition. When  $\Delta G$  is very low, a tiebreak hyper parameter ( $\tau$ ) is inserted to assist tree construction. This is accomplished by determining if  $\Delta G < \varepsilon < \tau$  is true while neglecting the  $\varepsilon$  condition  $\Delta G > \varepsilon$  (da Costa et al., 2018). Because the two highest

features have extremely comparable information gain values, the split can take place on either of them (Yang & Fong, 2011; Masrani et al., 2021).

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

where

$n$  : number of independent observations of real-valued random variable  $r$ .

$r$  : random variable whose range is  $R$ .

$1 - \delta$  : Confidence level, where  $\delta$  is the defined user.

$\delta$  : is one minus the probability of selecting the correct attribute at each node.

When the parameters of the algorithm change, evaluation metrics like accuracy and energy change as well, as it will be explained in the practical part. The parameters that will be altered include, split criterion, and bad attributes elimination. The  $n$  min parameter defines the minimum number of examples the algorithm must watch before computing  $\Delta G < \varepsilon$ ,  $\Delta G < \tau$ , but  $\varepsilon > \tau$  if there are sufficient statistics for a good split; the default value is 200. The parameter  $\tau$  identifies the parameter that will be utilized to break a tie in the event of a tie. When the difference between the two features is small enough, it suggests that both are equally good, therefore waiting a long time for more instances to make a split is pointless. The  $\delta$  parameter denotes one minus the likelihood of selecting the correct feature to be split. The researchers' default split criterion is information gain, and the Gini index was also tested. The last parameter to be changed is the removal of poor attributes (RPA). This parameter examines feature performance in order to identify features that perform badly and are unlikely to be picked for further splitting (García-Martín et al., 2017a).

### 2.2 Related Work

A lot of research work was done into improving the VFDT in various ways. A Moderated VFDT (M-VFDT) was proposed by Yang & Fong (2011), which uses an automated tie threshold to manage node splitting. The tree building approach is as quick as the native VFDT. The research presented two pre-pruning strategies for stream

mining, namely stringent and loose pruning, to deal with the exponential growth of tree size. Experiments are carried out to test the efficacy of their innovative methods. The M-VFDT with the pruning mechanism consistently outperforms the native VFDT.

Kourtellis et al. (2016) introduced the Vertical Hoeffding Tree (VHT) as the first streaming technique for learning decision trees that may be used to conduct classification tasks on such massive data streams reaching at high rates. VHT distributes decision trees in a unique way by utilizing vertical parallelism (Kourtellis et al., 2016).

García-Martín et al. (2017a) highlighted energy usage as a crucial component to consider when analyzing and testing data mining algorithms. This is backed up by an academic and practical analysis that shows how to construct efficient algorithms (García-Martín et al., 2017a). The same team looked at energy usage and tried to figure out why it was so high. They also conducted a fine-grained examination of the functions that utilized the most energy, helping in the understanding of that usage (García-Martín et al., 2017b). They also offered automated parameter adaptation in a different study to trade-off energy savings versus accuracy over runtime, and they provided the  $n$  min adaptation method to enhance parameter adaptation in Hoeffding trees. This approach saves energy by dynamically altering the number of examples required to make a split (García-Martín et al., 2018).

The Strict VFDT (SVFDT), a new approach that relies on the VFDT, was suggested in the research made by da Costa et al. (2018). By avoiding unnecessary tree construction, the SVFDT approach minimizes memory usage while retaining competitive prediction performance. SVFDT has also a faster processing time in comparison with VFDT since it builds considerably more shallow trees.

Losing et al. (2018) improved the efficiency of the algorithm, which seeks to split every  $n$  min instances on a regular basis. They used local statistics to estimate the split-time, minimizing superfluous split-attempts which are frequently the most expensive part of the computation. In practice, the algorithm uses the class distributions

of previous split attempts to estimate the minimum number of instances until the  $\varepsilon$  is satisfied (Losing et al., 2018).

The research work of Jia (2020) presented a technique for reducing the impact of concept drift and network noise by integrating the VFDT categorization mining method with the sliding window technique. According to test results, the method can successfully improve the accuracy of flow data categorization mining.

### 3. Theoretical Hyper Model of Enhanced VFDT

The high energy consumption is one of the most important problems facing the conventional techniques of the original algorithm. On the basis of the previous specialized works (Garcia-Martin et al., 2018; Garcia-Martin et al., 2021) it was found that some scenarios consume the greatest amount of energy. Therefore, the model proposed in this paper aims to solve this problem by using dynamic parameters in these scenarios for the algorithm that increases the energy consumption. This study discusses a fresh upgraded copy of the VFDT that covers various scenarios and the optimal method minimizing the energy for each of them. The first scenario is that when  $\Delta G > \varepsilon$  and the second scenario is that when  $\Delta G < \varepsilon$  and that contains two different cases. The first case implies that  $\Delta G < \varepsilon$ , but  $\Delta G > \tau$  and the second one implies that  $\Delta G < \varepsilon$ ,  $\Delta G < \tau$ , but  $\varepsilon > \tau$  if  $n \min \leq n_i$ .

#### 3.1 E-VFDT with RPA Parameter

In the first scenario, if  $\Delta G > \varepsilon$ , the leaf is replaced with a node. In this case, RPA may be used to improve accuracy while lowering energy utilization, and so minimize some VFDT functions that require far more energy while maintaining the high accuracy by deleting only unnecessary computations (García-Martín et al., 2021).

RPA analyses feature performance in order to identify bad features that are unlikely to be chosen for further split. Because the information gain from all the features is evaluated in each split, the attribute  $X_i$  is discarded if its value is lower than the information gain of the best feature by more than  $\varepsilon$   $G(X_a) - G(X_i) \geq \varepsilon$ . This strategy

should level up accuracy while limiting the amount of computations, by decreasing the number of operations required to calculate the information gain for all attributes (García-Martín et al., 2017b).

### 3.2 E-VFDT with Adaptive Tie Threshold

The second scenario is when  $\Delta G < \varepsilon$ . In this situation, reducing energy usage without sacrificing accuracy involves resolving the source of greater energy expenditure in each case.

The first case in the second scenario implies that  $\Delta G < \varepsilon$  but  $\Delta G > \tau$ . It indicates that estimating the operations connected to carelessly splitting the method leads to a leveling up in the energy usage of the algorithm. To maintain the tree growth under control, the standard VFDT requires a user-defined tie threshold where a split would just be forced to break. The tree size extends significantly when more data is flowed, and the accuracy of the classifier declines. Thus, the perfect constant tie threshold value cannot be found. As a result, for node splitting control, a dynamic tie threshold has to be utilized.

The study uses a dynamic tie threshold equal to the mean of  $\varepsilon$  as seen in Equation 3, which is in charge of node splitting during the tree-building process. When fresh data is received, the  $\varepsilon$  mean is dynamically computed and updated. Because  $\varepsilon$  must be estimated in any situation, as demonstrated in the following equation, it costs only a few more resources and  $\tau$  is changed in accordance with the  $\varepsilon$  mean value.

$$\tau_k = \frac{\sum_{i=1}^K \varepsilon_i}{K} \Rightarrow \tau_{k+1} = \frac{(\sum_{i=1}^K \varepsilon_i) + \varepsilon_{k+1}}{K+1} \quad (3)$$

When  $\varepsilon$  is customized with the fresh data, the new  $\tau$  is refreshed. With this new method, a dynamic  $\tau$  whose value is no longer defined by a single default number adapts, instead, to the incoming examples and  $\varepsilon$  means. So, the energy is decreased without loss of accuracy as the experimental results will demonstrate. Algorithm 1 shows the pseudocode of the execution of the E-VFDT with the adaptive tie threshold.

#### Algorithm 1. The E-VFDT with adaptive tie threshold

##### Input:

$S$ : the stream of instances  
 $\varepsilon$ : Hoeffding bound (HB)  
 $\delta$ : the error probability  
 $HT$ : Tree with a single leaf (the root)  
 $X, G(\cdot)$ : set of attributes  
 $G(\cdot)$ : split evaluation function  
 $\tau$ : the tiebreak parameter set by the user  
 $\tau_k$ : adaptive tie threshold

##### Output:

Enhanced Very Fast Decision Tree

##### Procedure:

1. Begin
2. While stream is not empty do
3. Read instance  $I_i$  from  $S$
4. Sort  $I_i$  to corresponding leaf  $L$  using  $HT$
5. Update statistics at leaf  $L$
6. Increment  $n_i$  instances seen at leaf  $L$
7.     if  $n_{\min} \leq n$  then
8.         Compute  $G(X_i)$  for each attribute  $X_i$
9.         Calculate  $\Delta G(\cdot) \leftarrow G(X_a) - G(X_b)$
10.         Compute  $\varepsilon$
11.         Compute  $sum \varepsilon(I) \leftarrow sum \varepsilon(I) + \varepsilon$
12.         Compute  $Mean \varepsilon(I) \leftarrow sum \varepsilon(I) / Count(I)$
13.         If  $\Delta G > \varepsilon$  OR  $\Delta G \leq Mean \varepsilon(I)$  then
14.             Split on best attribute  $X_a$  and Replace  $L$  with a node.
15.             for each branch of the split do
16.                 Update new leaves  
                     New leaf  $L_m$  with initialized statistics
17.                 End for
18.             Else If  $\Delta G < \varepsilon$  but  $\Delta G > \tau$  then
19.                 Calculate  $\tau_k$  corresponding to the value of mean  $\varepsilon$
20.                 End if
21.             End if
22.             Else
23.                 Do not split
24.                 Do not update HT
25.             End if
26. End while
27. End procedure

### 3.3 E-VFDT with Adaptive $n_{\min}$

The second case of the second scenario implies that  $\Delta G < \varepsilon, \Delta G < \tau$  but  $\varepsilon > \tau$ . Despite the reality that DSM algorithms adjust the modeling approach based on the observed examples, the parameters of these algorithms are fixed from the beginning. Fixed parameters can make an

algorithm to work improperly, thus utilizing extra computational resources and contributing to rise energy usage. Dynamic parameter adaptation for data stream mining algorithms is described in upgraded E-VFDT to trade-off energy efficiency versus accuracy throughout runtime. As previously stated, the algorithm needs to observe the  $n \min$  number of instances before calculating  $\varepsilon$  in order to determine if there are enough statics for a successful split. Because  $n \min$  instances are insufficient to create a reliable split in this situation, it is crucial to wait for further instances to decline  $\varepsilon$  and become lower than  $\tau$ . In an opposite way, when  $n \min$  is too high, the split attempt duration is prolonged and it is more probably to miss the optimal splitting time, leading to a long split-delay that slows the growth of the tree and degrades accuracy (Sun et al., 2020).

Computing the procedures relevant to splitting in an improperly way leads to level up the level energy of the algorithm. In this situation, adaptive  $n \min$  is required to degrade energy usage as a fixed value of  $n \min$  indicating that VFDT is not changeable to the properties of the data. It implies that  $n \min$  is the same throughout the flow of data (García-Martín et al., 2018).

To clarify the established way, the  $n \min$  adaptation method which is calculated in Equation 4 is utilized in order to improve the adaptation of the parameter in Hoeffding trees. Hoeffding tree algorithms figure out if a node seems to have enough  $n \min$  instances to create a reliable split. Based on those  $n \min$  instances, they estimate the information gain of the features and see whether there is a feature that obtains the most information gain.

However, because the  $n \min$  is fixed, if such examples are inadequate to accomplish a split, the algorithm wastes a lot of energy on such processes. The idea behind  $n \min$  adaptation is to adjust the number of  $n \min$  depending on the observed data, guaranteeing that the algorithm computes the best features only if there is a split to degrade energy usage and run time by relying on the setting of  $n \min$  in this scenario. This reduces the processing required to calculate all of the information gain of the features, cutting energy usage (Garcia-Martin et al., 2018).

$$Adaptive\ n\ min = \left\lceil \frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2\tau^2} \right\rceil \quad (4)$$

The number of examples required for a tie split is controlled by a parameter  $\tau$  which is defined through the  $\varepsilon$ . When  $\Delta G < \varepsilon < \tau$ , a tie-split is executed. In other words, after Adaptive  $n \min$  instances, a tie-split is conducted. Specifically, the default parameter value is utilized to make sure that the number of  $n \min$  watched by the algorithm is sufficient to generate a split with no energy loss, as the empirical results will demonstrate (Losing et al., 2018). In Algorithm 2 a pseudocode depicting the implementation of the E-VFDT with adaptive  $n \min$  is displayed.

**Algorithm 2.** The E-VFDT with a dynamic  $n \min$

**Input:**

$S$  : the stream of instances  
 $\varepsilon$  : Hoeffding bound  
 $\delta$  : the error probability  
 $HT$  : tree with a single leaf (the root)  
 $X$  : set of attributes  
 $G(\cdot)$  : split evaluation function  
 $\tau$  : the tiebreak parameter set by the user

**Output:**

Enhanced Very Fast Decision Tree

**Procedure:**

1. Begin
2. While stream is not empty do
3. Read instance from  $S$
4. Sort  $I_i$  to corresponding leaf  $L$  using  $HT$
5. Update statistics at leaf  $L$
6. Increment  $n_i$  instances seen at leaf  $L$
7.     If  $n \min \leq n_i$  then
8.         Compute  $G(X_i)$  for each attribute  $X_i$
9.         Calculate  $\Delta G(i) \leftarrow G(X_a) - G(X_b^i)$
10.         Compute  $\varepsilon$
11.         If  $(\Delta G > \varepsilon)$  or  $(\varepsilon < \tau)$  then
12.             Split on best attribute  $X_a$  and  
Replace  $L$  with a node.
13.             for each branch of the split do
14.                 Update new leaves,  
New leaf  $L_m$  with initialized  
statistics
15.             End for
16.         Else If  $\Delta G < \varepsilon, \Delta G < \tau$  but  $\varepsilon > \tau$  then
17.             Calculate adaptive  $n \min$
18.         End if
19.     End if
20.     Else
21.         Do not split
22.         Do not update HT
23.     End if
24. End while
25. End procedure

## 4. Experiments and Results

### 4.1 Datasets

To conduct this experiment, six datasets were obtained, three synthetic datasets generated with Massive Online Analysis (MOA) (Domingos & Hulten, 2000) and three real-world benchmark datasets from UCI (UCI Machine Learning Repository, n.d.). The present paper aims to test the proposed algorithm by using datasets with different characteristic, because E-VFDT is sensitive to data changes.

First dataset is real-world dataset. This dataset has been selected since it is the same one that the VFDT researchers utilize in their tests, therefore it serves as a basis for usual performance of the algorithm. The dataset contains examples attempting to forecast decent poker cards relied on a particular card. The other two datasets consist of real-world benchmarks, The first one is Forest Cover Type dataset and includes Predicting Forest Cover type from cartographic variables such as elevation, slope, and soil type. The second one is Spam dataset and determines whether an email is spam or not. The term “spam” refers to promotions for products and websites.

The first synthetic dataset is the hyper plane dataset, it uses a function to generate data that follows a plane in several dimensions. This dataset is often used to test algorithms that can handle the “drift” concept. The random tree dataset is the second synthetic dataset that was created using MOA. The synthetic generator generated one

million cases. The last synthetic dataset is the waveform dataset and involves the “drift” concept and noise.

The function generates a wave as a combination of two or three basic waves. The task is to differentiate between the three waves. Their main characteristics can be seen in Table 1.

### 4.2 Tool

For these experiments, MOA (massive online analysis) has been utilized as a framework. It is also used to operate the VFDT with various parameter configurations, and it’s dependent on the WEKA libraries as well. The MOA framework runs in tandem with IPPET (Intel(R) Platform Power Estimation Tool), a tool that can estimate how many power multiple processes to use (Zheng et al., 2019; Joshi & Patel, 2018).

The environment for practical application in this research also includes a set of specifications for the device used in this experiment, which in turn affects energy-related calculations such as the operating system: Windows 7 professional 64-bit (6.1, Build 7601), the processor: Intel(R) core (TM) i3-2350M CPU @ 2.30GHz (4 CPUs), 2.3GHz, and memory: 4096MB RAM.

### 4.3 Parameter Choice

Table 2 depicts a summary of the parameter setting, the parameter configurations include index of the VFDT (IDX),  $n$  min ,  $\tau$  ,  $\delta$  , Split criterion (S.CRT) which contains information Gain (S1) and Gini index (S2), and Removing Poor Attributes (RPA). The parameters  $n$  min , S.CRT, and RPA were ranged from the default option to the maximum

**Table 1.** Summary of the datasets used in the experiment

Datasets	Name	Type	Instances	Numeric features	Binary features
1	poker	Real world	1,025,010	5	6
2	Cover Type	Real world benchmark	581,012	10	44
3	spam	Real world benchmark	9,324	0	39,917
4	Hyper plane	Synthetic	1,000,000	9	1
5	random tree	Synthetic	1,000,000	5	5
6	waveform	Synthetic	60000	21	0

setting. Information Gain was evaluated by comparing it to the Gini index. Finally, the parameter for removing poor attributes was operated and eliminated to see how it performed.

**Table 2.** Parameter configuration index

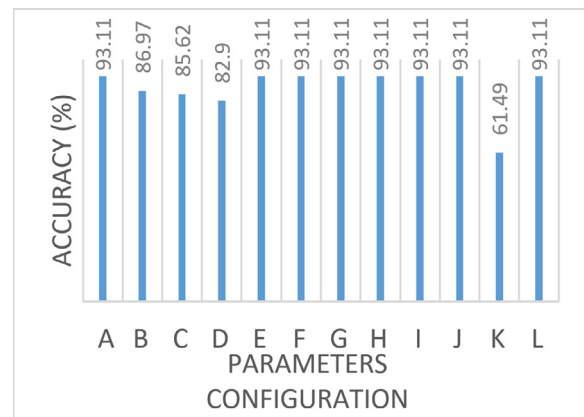
IDX	$n$ min	$\tau$	$\delta$	S.CRT	RPA
A	200	0.05	$10^{-7}$	S1	NO
B	700	0.05	$10^{-7}$	S1	NO
C	1200	0.05	$10^{-7}$	S1	NO
D	1700	0.05	$10^{-7}$	S1	NO
E	200	0.01	$10^{-7}$	S1	NO
F	200	0.09	$10^{-7}$	S1	NO
G	200	0.13	$10^{-7}$	S1	NO
H	200	0.05	$10^{-1}$	S1	NO
I	200	0.05	$10^{-4}$	S1	NO
J	200	0.05	$10^{-10}$	S1	NO
K	200	0.05	$10^{-7}$	S2	NO
L	200	0.05	$10^{-7}$	S1	YES

### 4.4 Empirical Study

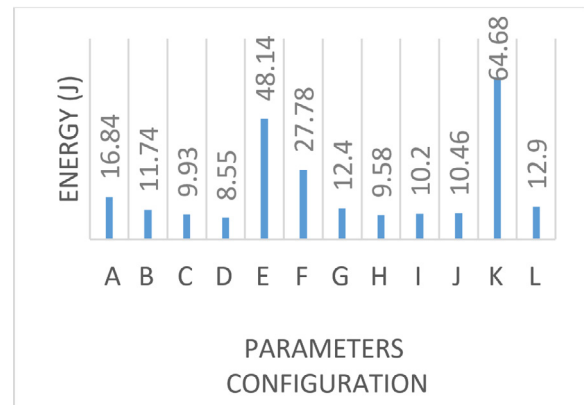
Figure 1 and 2 provide the findings of the experiment for poker dataset and estimated energy and accuracy. The effect of  $n$  min parameter across dataset can also be noticed. This parameter has no appreciable impact on the accuracy in experimental study.

However, a significant decrease in watts can be detected. The drop in power as  $n$  min rises is due to the fact that the value of  $\Delta G$  is calculated fewer times, hence conserving power. When it comes to energy as  $n$  min rises, energy falls. Lastly, as  $n$  min levels up, time will be reduced in general and it is sensible because there are fewer data sets and, thus, the tree is constructed faster. The accuracy does not change as the value of  $n$  goes up. Because of the reductions in time and power, there is energy reduction. The likelihood of making a correct split rises as  $\delta$  decreases. In empirical study, accuracy does not vary and the energy increases in a limited way due to the limitation of the Time Change and Power Change.

Whenever the split criterion is adjusted to the Gini index, the accuracy reduces significantly. According to energy, the rise in power and time has resulted in a rise in energy. When the parameter RPA is activated, better accuracy and decreased energy use can be predicted. Based on this experiment, it can be seen that accuracy is maintained throughout all datasets, while energy is reduced across dataset, owing primarily to a minor fall in power. It's worth noting that energy levels clearly change in different configurations. As seen in Figures 1 and 2 energy and accuracy change for each parameter in the dataset, emphasizing the importance of measuring energy usage.



**Figure 1.** How accuracy varies for the different parameter configurations

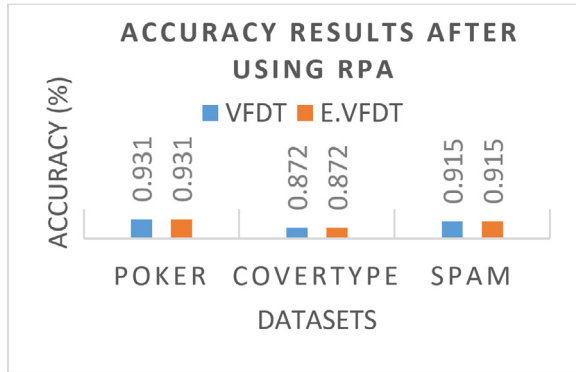


**Figure 2.** How energy varies for the different parameter configurations

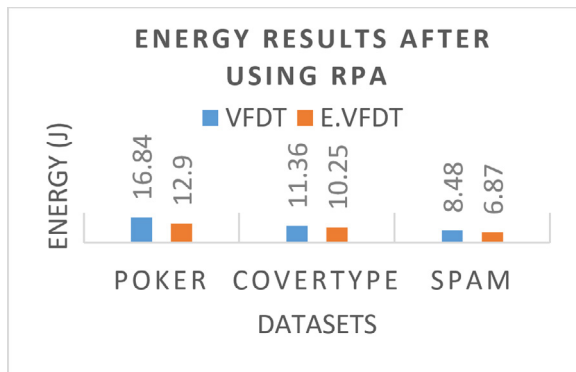
### 4.5 Experimental Design

The model focuses on VFDT's fundamental processes and determines which portions of the algorithm consume the most energy. When the RPA parameter is utilized in the first situation in the experimental findings, accuracy is maintained, while energy levels are reduced.

Figure 3 and 4 illustrate the findings of the experiment for each dataset, covering energy and accuracy. As it can be noticed, there is a significant variation after utilizing the RPA parameter, with the greatest performances of accuracy and energy for each dataset. This happens due to canceling the functions used to choose the appropriate attribute of splitting as these functions consume more energy.

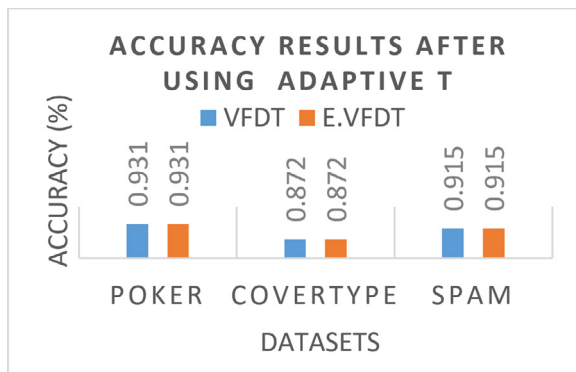


**Figure 3.** How accuracy varies when using RPA parameter

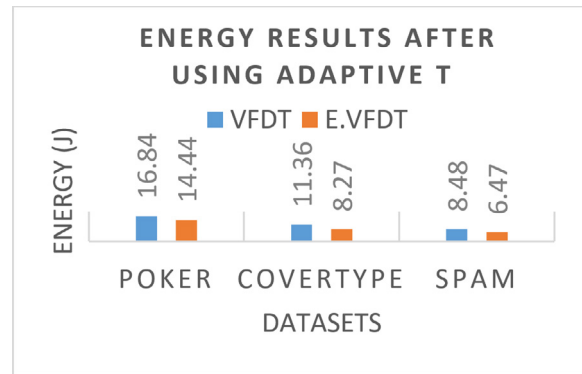


**Figure 4.** How energy varies when using RPA parameter

In the second scenario implies that  $\Delta G < \varepsilon$ , the first case if  $\Delta G < \varepsilon$  but  $\Delta G > \tau$ . Figures 5 and 6 illustrate the findings of the experiment for each dataset where there is a significant variation after utilizing adaptive  $\tau$ .



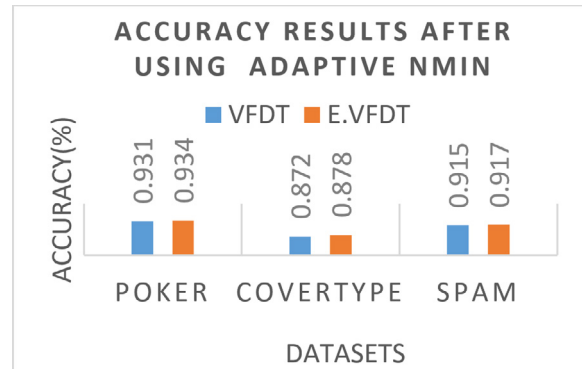
**Figure 5.** How accuracy varies when using adaptive  $\tau$



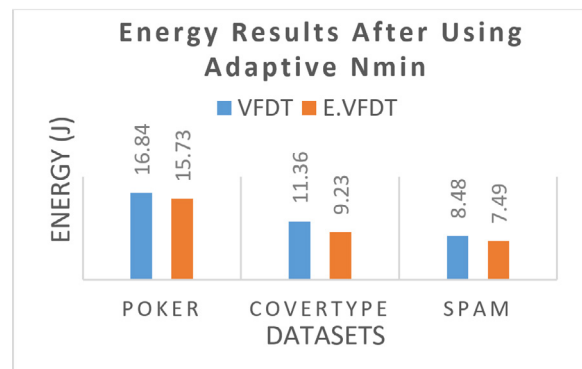
**Figure 6.** How energy varies when using adaptive  $\tau$

The findings demonstrate that the energy levels of the E-VFDT are lower than the energy levels of the original VFDT, while the means of the accuracy of the both algorithms are the same because the value of adaptive  $\tau$  varies according to the sequence examples and to the mean of  $\varepsilon$ . Accordingly, there is no waste in the energy level if its value is kept fixed.

In the second scenario if  $\Delta G < \varepsilon$ , the second case implies that  $\Delta G < \varepsilon, \Delta G < \tau$  but  $\varepsilon > \tau$ . Compared with the previous algorithm, after using Adaptive  $n$  min instead of fixed  $n$  min algorithm the energy consumption can be reduced and the accuracy can be improved to a certain extent as shown in Figures 7 and 8.



**Figure 7.** How accuracy varies when using adaptive  $n$  min



**Figure 8.** How energy varies when using adaptive  $n$  min



The results reveal that there is a statistical difference in energy levels and accuracy between the VFDT and the E-VFDT, while both of them are upgraded in the suggested method. *Adaptive n min* is based on the data regardless of the beginning *n min* so the algorithm will not calculate the gain for all the attributes unless there is a sufficient number of examples for the split to occur.

As far as is currently known, there are no previous works that have improved the performance of the VFDT algorithm in terms of energy consumption except the reference (García-Martín et al., 2018). in the  $\varepsilon$  equation in order to ensure that  $\Delta G \geq \varepsilon$  is satisfied during the next iterations, resulting in a split.

$$n \min = \left\lceil \frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2\Delta G^2} \right\rceil \quad (5)$$

The following is a comparative study of the practical experiment between the performance of the present new method and different scenarios between this previous work and 6 different datasets. This paper compares the VFDT standard algorithm with the P-VFDT modification in algorithm (Garcia-Martin et al., 2018), and with the E-VFDT enhanced algorithm.

The results of the experiments are divided in two sections. The first section presents the accuracy and energy consumption results of the real datasets setups, and the second one presents the results of the artificial dataset, respectively.

#### 4.5.1 Evaluation of the Results of Real-World Datasets

The results illustrated in Figures 9 and 10 show that the E-VFDT and P-VFDT (García-Martín et al., 2018) can consume less energy in comparison with the standard VFDT. The accuracy was also impacted but not greatly when compared to the amount of energy saved in the E-VFDT algorithm. There is a decrease of less than 1% in poker and spam datasets. It has been observed that the E-VFDT has consumed more energy than the P-VFDT (García-Martín et al., 2018) for poker and spam datasets but it consumed less energy for cover type dataset. Although, in all the three datasets, the E-VFDT is better than the P-VFDT (García-Martín et al., 2018) in the extent of its impact on accuracy

and this effect on accuracy is considered one of the defects of this previous work.

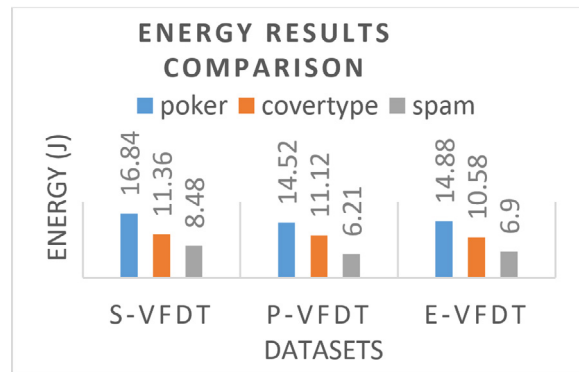


Figure 9. Results of the energy for each algorithm

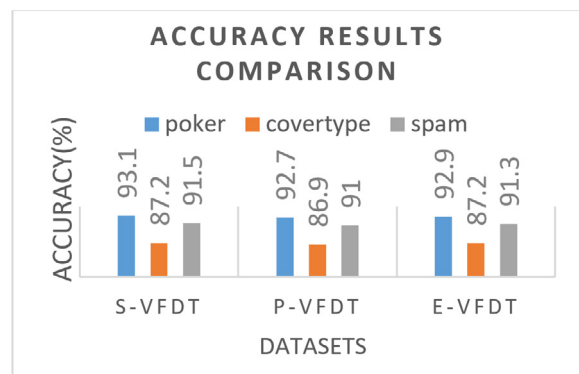


Figure 10. Results of the accuracy for each algorithm

#### 4.5.2 Evaluation of the Results of Artificial Datasets

E-VFDT significantly obtains higher energy consumption while it is not giving improvements in accuracy, but it obtains a lower accuracy than other algorithms for the hyper plane dataset as shown in Figures 11 and 12. The reason behind this result is the amount of drift on this dataset. Since the proposed algorithm is not able to learn the data with drift, there is no possibility for a higher accuracy in these dataset types, which is considered an imperfection of the present algorithm.

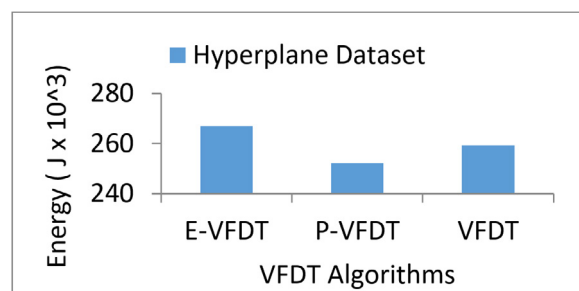
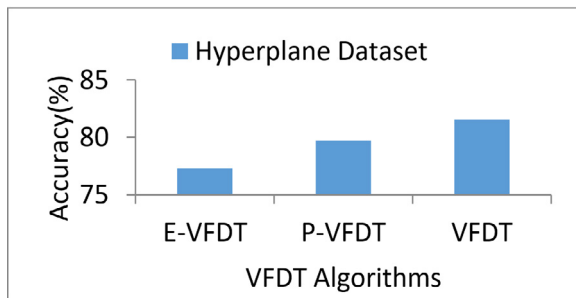
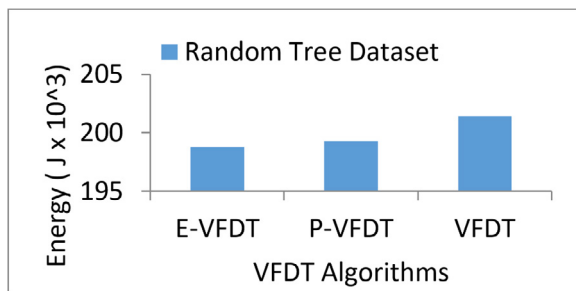


Figure 11. Results of the energy for each algorithm in hyper plane dataset

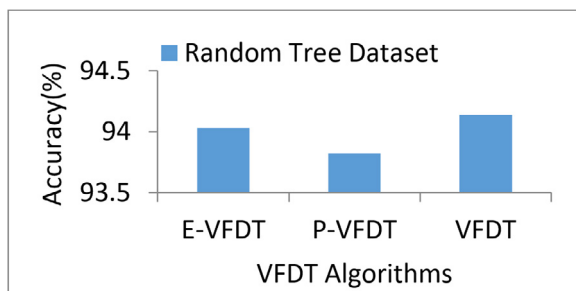


**Figure 12.** Results of the accuracy for each algorithm in hyper plane dataset

In Figures 13 and 14 it can be observed that the E-VFDT obtains the lowest energy consumption in random tree dataset, despite it contains a large number of instances. The reason is that the proposed algorithm is able to adapt  $n$  min based on the data regardless of the beginning  $n$  min, so the algorithm will not calculate the information gain for all attributes unless there are sufficient examples for the split to occur. Moreover, E-VFDT scales better than P-VFDT in terms of accuracy.



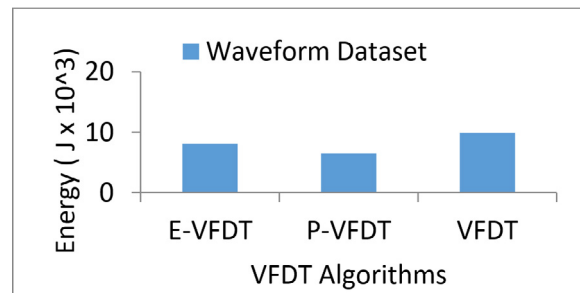
**Figure 13.** Results of the energy for each algorithm in random tree dataset



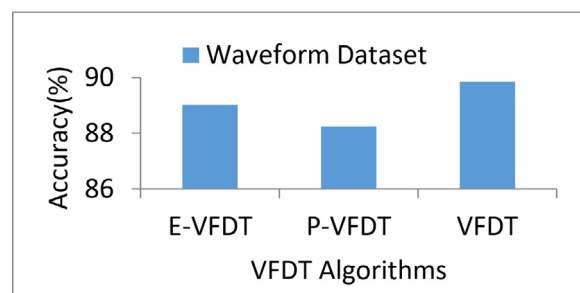
**Figure 14.** Results of the accuracy for each algorithm in random tree dataset

As it can be observed from Figures 15 and 16, the E-VFDT and P-VFDT obtain lower energy consumption in comparison with the VFDT in waveform dataset, but the E-VFDT consumes more energy than the P-VFDT. The reason behind this result is the amount of noise on this dataset. Since the proposed algorithm is not able to learn the data with noise, especially on a small number of

dataset instances, because it depends on an adapted  $n$  min, there is no possibility for a higher accuracy in these dataset types. here is no possibility for a higher accuracy in these dataset types.



**Figure 15.** Results of the energy for each algorithm in waveform dataset



**Figure 16.** Results of the accuracy for each algorithm in waveform dataset

The obtained results show that the E-VFDT achieves a much better performance in comparison with the standard VFDT and P-VFDT (García-Martín et al., 2018) in reducing the amount of unnecessary energy, by providing several solutions to the causes of its high value. The first solution is to use RPA parameter, because the information gain of all features is evaluated in each split, RPA analyses feature performance to identify poor features that are doubtful to be picked for further split, and thus eliminate the functions that consume more energy. The second solution is to use a dynamic  $\tau$  whose value changes depending on the incoming examples and mean  $\varepsilon$  rather than being defined by a default value. Therefore, it is not possible to know in advance which value of constant tie threshold is the best one. As a result, energy is conserved while accuracy is preserved. The last solution is to use adaptive, the goal of  $n$  min adaptation is to update the number of  $n$  min based on the observed instances, ensuring that the algorithm computes only the best features if there is a split to save energy consumption.

It should be noted that the total computational complexity of the new algorithm did not exceed

its value in the original algorithm, since it was Big  $O(n)$  and still retains the same value after modification.

## Conclusions

The present work introduces a hyper method to enhance the VFDT, which enables for an energy-efficient strategy to build Hoeffding trees without hurting their predictive effectiveness, resulting in lower energy usage and negligible accuracy loss. The accuracy and energy levels of the original algorithm and of the enhanced one are compared to those of the ordinary VFDT. According to the findings, E-VFDT requires less energy than VFDT for dataset types that contain a huge number of examples. On the contrary, E-VFDT is not giving improvements in energy for datasets with drift. thus, being unable to learn the data with

drift. An investigation was done in particular to see how the conventional algorithms behaved in terms of energy usage and accuracy when certain parameters were changed. The findings of experiments made within this study show that it is possible to reduce the energy consumption of the algorithm without sacrificing accuracy by correctly altering the parameters of the algorithm.

Finally, the mentioned algorithms are compared in different datasets. The E-VFDT used less energy than the conventional VFDT and was more accurate than it.

Further methods are offered for future work in order to enable for an energy-efficient approach to designing Hoeffding trees without compromising their predictive effectiveness especially in datasets which have noise and drift.

## REFERENCES

- da Costa, V. G. T., Carvalho, A. C. & Junior, S. B. (2018). Strict Very Fast Decision Tree: a memory conservative algorithm for data stream mining, *Pattern Recognition Letters*, 116, 22-28.
- Domingos, P. & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the 6th SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 71-80).
- Fong, S., Song, W., Wong, R., Bhatt, C. & Korzun, D. (2018). Framework of Temporal Data Stream Mining by Using Incrementally Optimized Very Fast Decision Forest, *Internet of Things and Big Data Analytics Toward Next-Generation Intelligence*. 483-502. Springer, Cham.
- García-Martín, E. (2017). Energy efficiency in machine learning: A position paper. In *30th Annual Workshop of the Swedish Artificial Intelligence Society SAIS, Vol. 137* (pp. 68-72). Karlskrona. Linköping University Electronic Press.
- García-Martín, E., Lavesson, N. & Grahn, H. (2017a). Energy efficiency analysis of the very fast decision tree algorithm, *Trends in Social Network Analysis*, 229-252.
- García-Martín, E., Lavesson, N. & Grahn, H. (2017b). Identification of energy hotspots: A case study of the very fast decision tree. In *International Conference on Green, Pervasive, and Cloud Computing* (pp. 267-281). Springer, Cham.
- García-Martín, E., Lavesson, N. & Grahn, H. (2021). Energy-aware very fast decision tree, *International Journal of Data Science and Analytics*, 11(2), 105-126.
- García-Martín, E., Lavesson, N., Grahn, H., Casalicchio, E. & Boeva, V. (2018). Hoeffding Trees with Nmin Adaptation. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, (pp. 70-79).
- Ibrahim, A. A. & Abdulaziz, R. O. (2020). Analysis of Titanic Disaster using Machine Learning Algorithms, *Engineering Letters*, 28(4), 1161-1167.
- Jia, S. (2020). A VFDT algorithm optimization and application thereof in data stream classification, *Journal of Physics: Conference Series*, 1629(1), 1-8. IOP Publishing.
- Joshi, A. P. & Patel, B. V. (2018). Comparative Study of Different Classification Algorithms for Stream Data Mining Using MOA, *International Journal of Computer Sciences and Engineering*, 6, 7-11.
- Kok, S. H., Abdullah, A. B., Supramaniam, M., Pillai, T. R. & Hashem, I. A. (2019). A Comparison of Various Machine Learning Algorithms in a Distributed Denial of Service Intrusion, *International Journal of Engineering Research & Technology*, 12(1), 1-7.
- Kosina, P. & Gama, J. (2013). Very fast decision rules for classification in data streams, *Data Mining and Knowledge Discovery*, 29(1), 168-202.
- Kourtellis, N., De Francisci Morales, G., Bifet, A. & Murdopo, A. (2016). VHT: Vertical hoeffding tree. In *2016 IEEE International Conference on Big Data (Big Data)*, (pp. 915-922). DOI: 10.1109/BigData.2016.7840687

- Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J. & Wozniak, M. (2017). Ensemble learning for data stream analysis: A survey, *Information Fusion*, 37, 132-156.
- Losing, V., Wersing, H. & Barbara Hammer, B. (2018). Enhancing very fast decision trees with local split-time predictions. In *2018 IEEE International Conference on Data Mining (ICDM)*, (pp. 287-296).
- Masrani, A., Shukla, M. & Makadiya, K. (2021). Empirical Analysis of Classification Algorithms in Data Stream Mining. In *International Conference on Innovative Computing and Communications* (pp. 657-669). Springer, Singapore.
- Prasad, B. R. & Agarwal, S. (2016). Stream Data Mining: Platforms, Algorithms, Performance Evaluators and Research Trends, *International Journal of Database Theory and Application*, 9(9), 201-218.
- Sun, J., Jia, H., Hu, B., Huang, X., Zhang, H., Wan, H. & Zhao, X. (2020). Speeding up very fast decision tree with low computational cost. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, (pp. 1272-1278).
- UCI Machine Learning Repository (n.d.). *Datasets*. Available at: <<https://archive.ics.uci.edu/ml/datasets.php>> last accessed: 1<sup>st</sup> of January, 2022.
- Yang, H. & Fong, S. (2011). Moderated VFDT in stream mining using adaptive tie threshold and incremental pruning. In *International Conference on Data Warehousing and Knowledge Discovery*, (pp. 471-483). Springer, Berlin, Heidelberg.
- Zheng, X., Li, P., Chu, Z. & Hu, X. (2019). A Survey on Multi-Label Data Stream Classification, *IEEE Access*, 8, 1249-1275.