Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7627747

# LINEAR PREFIX TREE AND LAPLACE ACCURACY BASED FREQUENT PATTERN MINING TECHNIQUE

**J.Gayathri**

M.Sc .,M.Phil, Ph.D Research Scholar, Department of Computer science, Kongunadu Arts and Science College, Coimbatore, Tamil Nadu, India, kavimsc07@gmail.com

**Dr.S.Mythili**

MCA .,M.Phil., Ph.D, Associate Professor and Head, Department of Information technology, Kongunadu Arts and Science College, Coimbatore, Tamil Nadu, India
smythili87@gmail.com

**Abstract:** Frequent Pattern Mining (FPM) is the most focused research by numerous researchers to predict the repeated activities happening on different sectors. The FPM is done utilizing Lasso Regression based Improved Frequent Pattern Mining Detection Scheme (LR-IFPMDS) in our previous research. However, in the existing work, performance degradation occurs at the time of attribute selection with multiple objectives. It reduced the overall prediction accuracy. Therefore, this work focussed on to overcome the existing work limitation and introduced Linear Prefix Tree based Frequent Pattern Mining Detection Scheme (LPT-FPMDS) for mining frequent patterns. In this work initially, optimal attribute selection performed using Hybrid crow swarm Optimization with Cat Swarm Algorithm. Here fitness values considered are Accuracy, Error rate, Information Gain and Gain Ratio. Based on selected attributes frequent pattern rule generation is performed. Rule pruning process is executed utilizing Laplace accuracy. In this work, rule pruning and frequent pattern mining scheme is combined together to obtain the frequent patterns. The proposed FPM technique uses Linear Prefix Tree (LPT) for recognizing the frequent patterns. In LPT, once the leaf node reached, it estimates the Laplace accuracy and executes the rule pruning procedure based on the estimation outcome. This enhances the accuracy and curtails the computation overhead. The entire work is executed using Matlab simulation tool. The experimentation result shows that the proposed model has performance improvement in terms of accuracy.

**Keywords:** Pattern rules, frequent pattern, Linear Prefix Tree, Laplace accuracy, leaf node of tree.

## 1. Introduction

Data mining is the method involved with uncovering non-insignificant and possibly helpful data from enormous data sets [1]. Affiliation Rule Mining (ARM) is a significant information mining method that is used for finding the examples/rules among things in a huge data set [2]. The objective of ARM is to recognize gathering of things, which happen together, for instance in a market container investigation. Mining affiliation rules contains two stages: the first is producing incessant itemsets [3]. The second is producing affiliation rules. The fundamental test in affiliation rule mining is to recognize incessant itemsets. Finding continuous itemset is one of the significant stages in affiliation rule mining [4]. Since the arrangement of second sub-issue is straight forward, the greater part of the analysts had center around how to produce regular itemsets.

Frequent itemsets are the item sets, which happen oftentimes in the exchange data set [5]. The goal of Frequent Itemset Mining is to distinguish all the regular itemsets in an exchange information base. Additionally, the high and low benefit esteems are determined with the selling frequencies [6]. For instance, some as often as possible sold things, for example, bread, pen and milk might have lower benefit esteems when contrasted with that of rarely sold higher benefit esteem things like gold, platinum and jewel. In this manner, observing just customary successive examples in an information base can't satisfy the prerequisite of finding the most significant itemsets/clients that add to the significant piece of the absolute benefits in reality retail data set [7]. This inspired to foster a mining model to find the itemsets/clients adding to most of the benefit [8]. A continuous itemset is the itemset having recurrence support more noteworthy a base client determined threshold.

The high utility itemset mining issue finds the itemsets having higher utility than the client defined utility [9]. The utility of the itemset is defined as the worth or benefit associated with each thing in an information base [10]. For example, PC framework is more productive than phone as far as benefit. Utility characterize as Interestingness, productivity or significance of thing [11]. Utility estimated as far as cost benefit or other client inclination.

High utility mining are used in various applications such as cross showcasing in retail locations, site click stream examination, observing essential instances in biomedical applications and online business [12]. The high utility itemset mining techniques and its variations specified in the references mainly focussed to perform on exact datasets. An exact data set contains definite data about the situation with its exchanges. Nonetheless, there is one more sort of genuine world value-based datasets, called unsure data sets, on which high utility itemset mining strategy is proposed. A portion of the data might have been lost or not be accessible in dubious datasets. There are additionally various techniques, which remove high utility itemsets from dubious data sets.

The objective of this research is to develop a reliable Frequent Pattern Mining strategy to discover the frequent patterns with higher efficacy. This research work ensures the detection rate and accuracy of efficient Frequent Pattern Mining approach.

## 2. Literature Review

Vlasselaer et al [13] developed a model called APATE, for detecting the online transaction fraud using network-based extensions. In this approach, intrinsic and network based attributes extracted to recognize the online transaction frauds. The different classification models such as Logistic Regression (LR), Neural Network (NN) and Random Forest (RF) make use of the extracted attributes for detecting the fraudulent transactions. Among the classification model RF gained the Area under Curve (AUC) value of 0.986. However, this model discovers the individual frauds and it fails to detect the group behaviour.

Jha et al [14] deployed transaction aggregation strategy based on logistic regression to recognize the credit card frauds. In this approach, credit card transaction history was aggregated to determine the credit card frauds. The primary attributes and quantitative attributes from credit card transaction dataset are extracted for identifying the fraudulent behaviour. However, reliability of the model is not ensured while handling the complex datasets.

Leu et al [15] developed Secure M-Commerce System (SMCS) for secure credit card transactions. In this approach, trading system's cash flow and credit card entities are coordinated to enable secure transactions. In addition, this model uses Data Connection Core (DCC) that has the link between the bank and consumer for enhancing the security. SMCS uses two dimensional cipher technique for protecting the transmitted message and it has the ability to protect from different type of attacks during transaction through authentication process. This model is simulated on consumer merchant environment utilizing JAVA program and noted that it requires less key generation timing compared to existing models. However, efficacy of this model is low compared to the baseline models.

Zhang et al [16] portrayed the performance development is fraud detection model using Homogeneity-Oriented Behaviour Analysis (HOBA) mechanism. In this approach, HOBA features are extracted from dataset to detect the credit card frauds effectively. The Deep Belief Network (DBN) classifier with HOBA features achieved 95.25% of accuracy in fraud detection, which is higher compared to other models. The real time dataset of Chinese commercial bank is utilized for evaluation. However, it has complex structure.

Gianini et al [17] suggested the use of Coalitional Game Theory (CRT) within the power indexes and Shapely value for managing the pool of credit card fraud detection rules. In this approach, rules enabled in Near Real Time (NRT) classification phase are selected using CRT approach for enhancing the performance of fraud detection model. This model is evaluated using real world dataset and observed that one tenth of the rules produce notable performance than using the entire rules. However it is not suitable for analysing data with higher complexity.

Lucas et al [18] utilized multi-perspective Hidden Markov Model (HMM) for enhancing the performance of fraud detection model. In this approach, different perspectives of credit card transactions are considered for determining the frauds. The historical features of these perspectives are extracted utilizing HMM. These features are used by Random Forest (RF) model for recognizing the frauds. This model has the performance improvement up to 18.1% while examining the Belgian credit card transaction. However, it takes longer time for detection.

Jurgovsky et al [19] deployed Long Short-Term Memory (LSTM) technique for credit card fraud detection. In this system, feature engineering techniques such as time delta, feature aggregation are used to extract the features for improvising the detection accuracy in both online and offline status. Then, these features are used by LSTM for prediction. The effectiveness of this model is evaluated using real time fraud detection dataset and it is contrasted against RF model. However, RF model performs better in certain cases.

Bahnsen et al [20] used Von Mises Distribution (VMD) procedure for credit card fraud detection. In this model, periodic behaviour during transaction is analysed using VMD instead of aggregation method. The features determined are used by LR, RF and decision Tree (DT) models for identifying the credit card frauds. This model is examined using European credit card fraud dataset. The result analysis shows that the inclusion of periodic features in fraud detection increase the performance up to 13%. However, this model has time complexity.

Sa et al [21] recommended customized Bayesian Network Classifier (BNC) for credit card fraud detection. In this model, Hyper Heuristic Evolutionary Algorithm (HHEA) implemented to enhance the performance of BNC in credit card fraud detection. The experiments are conducted on the PagSeguro dataset. This model attained the efficacy up to 72.64%. However, it has high computational complexity.

## 3. LINEAR PREFIX TREE BASED HIGH FREQUENT PATTERN DETECTION

In this work, a hybrid technique, which is the combination of Crow swarm Optimization with Cat Swarm Algorithm, enabled for feature selection. Here the considered fitness values are accuracy, error rate, Information Gain and Gain Ratio. Based on selected attributes pattern rule generation is performed. Rule pruning procedure performed based on Laplace accuracy. This work integrates the rule pruning and pattern mining process. The proposed work uses linear prefix tree to detect the frequent patterns.

### 3.1. AUGUMENTED LAGRANGIAN BASED RANKING

Each metric have to measure with normalized discounted cumulative gain (NDCG) if the objective is to optimize multiple metrics on ranking. When the objectives are given, as upper bounds in the cost function then each objective must achieves the Œ minimum criteria. Specifically, this work directs to use the cost function used in λ-MART to NDCG and sets the UB on cost $bt$ (i.e., Ct (s) ≤ , t = 1, fi,T , with s being the predictive scores of the model). Typically, the cost of base line model UB is set as fraction (%) of cost. Therefore, the cost rescaled as per the modifications and hence UB become more intuitive. The cost reduction is achieved up to 10% when b has the value 0.9. The constraints used in this system are in the form of cost function. Hence, the constraint optimization problem is modelled as $\min_s C^{pm}(s)$ s.t. Ct (s) ≤ bt, t = 1, ...,T , pm : primary objective. With the dual variables α, AL at iteration k is written as follows:

$$L_k(s,\alpha) = C^{pm}(s) + \sum_{t}^{T} \alpha^t \left(C^t(s) - b^t\right) - \sum_{t}^{T} \frac{1}{2\mu}\left(\alpha^t - \alpha_{k-1}^t\right)^2$$

where $\alpha_{k-1}^t$ representing the solution of previous iteration and the constant used in current iteration is denoted as $k$. Here, μ is the large constant. In the above equation, the last term is the representation of augmented term, which produced proximal minimization with $\alpha_{k-1}^t$ iterations in order to convert the Lagrangian Optimization smooth.

### 3.1.1. HYBRID CROW SWARM AND CAT SWARM BASED ATTRIBUTE SELECTION

In this work, optimization algorithm specifically crow swarm with cat swarm calculation is used for the ideal property determination. The targets considered in this work for ideal property choice are precision, mistake rate, data gain, and gain proportion. The credits with greatest data gain, expanded precision, lesser blunder rate and least Gini record are picked for the standard age. Cat Swarm Optimization (CSO) calculation is essentially swarm canny calculation that draws its motivation from the conduct of felines. Crow Search (CS) is developmental calculation relies upon the conduct in groups of crows. Crows are the most intelligent birds. The crows notice the food concealing activity of other birds and it takes the food of that bird when it leaves the place. crows have good memory and it has the ability of recalling the concealing spots even after a couple of months. This memory force of crow search calculation is joins with the feline multitude calculation for the ideal trait determination. The computation carried out in the hybrid model is explained below.

Based on augmented Lagrangian score, ranking will be calculated and that will be considered as fitness value in the proposed hybrid swarm optimization algorithm. The calculation proposed in hybrid model for every $i$th individual(i=1,2,...,N) is described as follows.

1) At first, the movable boundaries and the problems are initialized. Boundaries are number of people (N), looking for memory pool (smp), count aspect to change (cdc), looking for range aspect (srd), speed increase coefficient (c), blending proportion (mr), flight length (fl), and mindfulness likelihood (AP). In this work, worth of flight length is liable for investigation and abuse process. When the flight length, fl $< 1$, the location of crow $i$ moves towards the location of crow j and it is resulted with neighbourhood search. In the situation, fl $> 1$ the crow i stay away from crow j and it enables the global search.

2) Next, memory, speed and the position of the population are initiated randomly. In the D layered space, N population are arbitrarily located. Where $X_{i,d}$ representing the individual $i$th individual's addressing position with the aspect d, $V_{i,d}$ representing the addressing speed and $m_{i,d}$ denotes the addressing memory. In the beginning stage, underlying situation is taken as the underlying memory.

3) The wellness (objective) capacity of every individual is evaluated.

4) If $X_{i,d}$ is identified as the best solution (xbest) then self-position consideration (SPC) is set as to one (SPC=1) for the i th individual else the SPC=0.

5) With respect to mr, the randomly selected population and its banner are set to looking for mode. The banner is set to the following mode.

6) The looking mode is applied to the population which requires the model utility clarification from past else the following model is executed.

7) The situations of new individuals are collected by considering the history of following and looking for mode.

8) The memory is updated when the wellness work worth of new position is superior to the wellness work worth of the retained position.

9) The new situations are generated based on the picked duplicate in looking for mode and its position based on following mode.

10) Then, the wellness (objective) capacities of every individual in the new position are evaluated.

11) The memory is updated when the wellness worth of the retained position is not adequate compared with the new position.

12) Check the end basis. When the requirement is satisfied, the best situation in memory is accounted as the best arrangement. In any case, rehash stages 4) – 12).

### 3.3. FREQUENT PATTERN MINING RULE GENERATION

After attribute selection, pattern mining rule generation is finished. A Frequent example rule is an affiliation decide that is checked when the normal rule happens more than once. Overall, it rises to the top when the prevailing impact delivered by the solid rule is eliminated. In this methodology, there is no burden over the help of the bizarre and the reference rules. Assume X and Y are the two non-void itemsets which are present in the thing D. % We characterize a successive example rule by the triple (Fcsr, Fanom, Fref) made out of three standards fulfilling the accompanying conditions:

$X \rightarrow Y$ is a frequent rule (Fcsr)

$-Y \rightarrow A$ is certain in DX % (Fanom)

$A \rightarrow -Y$ is certain in DX % (Fref)

It is proved that $Conf_x (A \rightarrow B) = Conf (X \wedge A \rightarrow B)$ and this is not appropriate in the case of using with certainty factor. The complication occurs due to the existence of consequent's support in DX or D in computing the certainty factor.

$$CF (X \wedge -Y \rightarrow X \wedge A) \neq CF_x (-Y \rightarrow A)$$

$$CF(X \wedge A \rightarrow X \wedge -Y) \neq CF_x (A \rightarrow -Y)$$

Because

$$Supp (X \wedge A) = \frac{|X \cap A|}{|D|} \neq \frac{|X \cap A|}{|X|} = supp_x(A)$$

### 3.4 RULE PRUNING USING LAPLACE ACCURACY

Later rule generation, rule pruning is done dependent on the measurement called Laplace exactness. Since there are huge number of rules, finding delegate rules is a troublesome assignment. Rather than handling every one of the (few a great many) affiliation governs, a couple thousands (say under half) of profoundly positioned rules might be utilized to track down agent rules Care should be given in fixing rate for choosing profoundly positioned rules, since determination of tiny level of rules may not cover all the exchange adequately. Intriguing quality measures can be utilized to rank the affiliation rules. The exceptionally positioned little level of rules is utilized to find agent rules for every exchange. These found principles of every exchange are contrasted and the agent decides that are found for the comparing exchange utilizing the entire arrangement of affiliation rules. This analysis is rehashed for every intriguing quality measure to rank and prune affiliation rules, to find delegate rules and to analyze. In this work Laplace exactness is used to characterize the intriguing quality of various number of rules.

Laplace exactness is for the most part utilized in grouping to gauge the normal mistake of a standard. This normal precision for a given rule, r, is given by the equation:

$$\text{Laplace (r)} = \frac{(p_c(r) + 1)}{(P_{tot}(r) + m)}$$

Where m is the quantity of classes in the area, $P_{tot}$ (r) is the quantity of occasions matching r precursor and $p_c$ (r) is the quantity of cases covered by r that have a place with class c.

When all guidelines are found, positioned and the classifier built, and an experiment (ts) is going to be anticipated, this work go over the standard set and denotes all principles in the classifier that might cover ts. Assuming that more than one rule is pertinent to ts, this work separates them into bunches as indicated by the classes, and ascertains the normal anticipated exactness for each gathering. Finally, the class with the biggest normal expected precision esteem is doled out to ts.

### 3.5 FREQUENT PATTERN MINING USING LINEAR PREFIX TREE

The LP-tree has accompanying design: (1) The primary design is the Header list which incorporates the supports, hub joins and the thing name (2) Linear Prefix Node (LPN) for putting away regular things of every exchange and a comparing header, and (3) Branch Node List (BNL), it contains the data of branch as well as their youngster hubs. LP-tree comprising of c LPNs.

$$\text{LP-tree} = \{\text{Headerlist, BNL, } LPN_1, LPN_2, \ldots, LPN_c\}$$

LP-tree altogether has a straight construction. Each set of incessant things is saved into hubs made out of a cluster structure, where we utilize numerous exhibits since one exhibit structure can't communicate things as a tree structure with many branches. To interface each exhibit, each cluster has a header in the initial segment of the cluster, where the header demonstrates its parent cluster. LPN contains a header and a cluster hub putting away example, and the exhibit hub comprises of a few interior hubs. Besides, the base of the tree is the header of any LPN which is embedded first in the tree.

The development of tree is explained as follows. We filter a data set and count all thing upholds. From there on, we sort all things in their backing plunging request and afterward produce a relating header list, where the rundown stores things as per the arranged request. In particular, the upper things in the rundown have more noteworthy backings while the lower ones have qualities that are more modest. The inclusion procedure enabled for LP tree exchange is defined with two cases. The previous one is that the principal exchange is embedded into the LP-tree. Its strategy is as per the following. To begin with, we create LP-tree by filtering the data set again and sort the main exchange contingent upon the arrangement of the header list. That is, its things with more modest help than the base help are erased, and the excess things are arranged in help plummeting request. From that point forward, we embed the arranged exchange into the tree, where LPN is made and associated with a root since the tree is at first unfilled. During this stage, the principal exchange is forwarded to one LPN and it has many inner exhibit hubs as like the exchange length. If the exchange length is n, then the things of exchange have to embed in one LPN and the size of LPN with header is $n + 1$.

The header of LPN is associated subsequently to its parents for embedding the exchange things. In which, the header is interlinked with root when the current LPN is added first to the tree. The pointer of the root is added to the branch hub table and it stores the initial hub of recent LPN into the youngster hub list associated with root pointer. The subsequent case is the point at which each of the exchanges with the exception of the first are included the tree. Its inclusion is preceded as follows. The rare things in the embedded exchange are eliminated and sorts the continuous things in help plunging request. Then, the current LPN's addresses, BNL of the root and main hub are added to make the kid hub by a root and the branches are created subsequently. In this stage, the exchange is embedded as a contrast to the root. From there on, we affirm all the kid hubs of the root with BNL data since the past exchange is as of now included the tree, for example, the root has at least one kid hubs, where, we at first really look at the interior youngster hub of the current LPN.

On the off chance that the thing to be embedded is equivalent to the thing of the actually look at hub (the inner hub), the current area moves to that hub, and its backing is expanded by 1. In any case, to affirm the other kid hubs, we read the relating branch data in BNL and afterward increment backing of the current thing by 1 assuming the thing is equivalent to the hub got from BNL. Assuming it isn't equivalent to that, we create another LPN and addition remaining things of the exchange in the new LPN, where the current hub turns into a branch hub and is included BNL. Expecting that n is the length of any exchange and r is the quantity of things previously embedded in the past LPN, we store the leftover things in the new LPN simultaneously, where the quantity of exhibit hubs in the LPN is n r + 1 (counting a header). To store all exchanges without any issues, LP-tree interfaces each of its hubs in one of two ways. In the first place, inward hubs of LPN are straightforwardly associated with one another with practically no pointer. Second, when any branch happens, LPtree joins relating youngster hubs using BNL. Handling everything exchanges, we can acquire a total LP-tree. When the LP-tree development ends, BNL is disposed of in light of the fact that it isn't utilized any more extended.

## 4. Results and Discussion

In this section, the proposed model of associative classifier is evaluated in MATLAB. The proposed Linear Prefix Tree based Frequent Pattern Mining Detection Scheme (LPT-FPMDS), previous work Sliding Window Based Tilted Time Frame (SWTTF) algorithm is compared with the existing Tilted frame based Frequent Pattern Mining (TFFPM) Algorithm.

### 4.1. Dataset Description

In this work, foreign exchange rates dataset is utilized. This dataset is obtained from the Federal Reserve's Download Data Program. Few modifications are performed in the dataset like inverting base currency and header simplifications. For example, Fed includes currencies of different countries such as Australian dollar, Euro, New Zealand Dollar and United Kingdom Pound based in their units (not in dollar). Therefore, the conversion is made to view all the currencies in one unit that is dollar. The tree structure of dataset is displayed in figure 1.

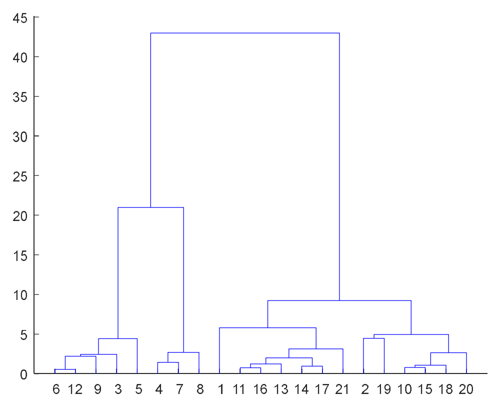


Figure 1. Tree structure of dataset

### 4.2. Numerical analysis

The performance metrics that are considered in this research method for the evaluation of the proposed and existing research method are "number of support data, time, frequent itemsets found and number of rules found". The number of support data exist are displayed in table 1.

Table 1. Number of support data comparison

| Minimum Support | Number of support data | | |
|---|---|---|---|
| | SWTTF | TFFPM | LPT-FPMDS |
| 0.1 | 6592 | 6783 | 6892 |
| 0.2 | 5680 | 5869 | 6026 |

| | | | |
|---|---|---|---|
| 0.3 | 5369 | 5502 | 5756 |
| 0.4 | 5307 | 5477 | 5612 |
| 0.5 | 5298 | 5417 | 5516 |
| 0.6 | 5287 | 5390 | 5498 |

The graphical comparison of the number of support data found is given in the following figure 3.

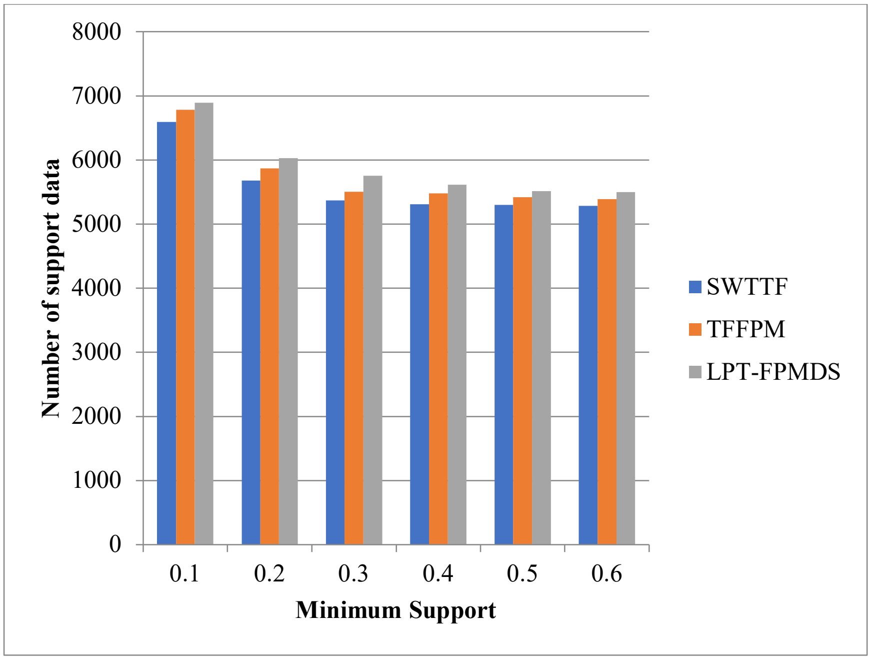


Figure 3. Number of support data comparison analysis

In figure 3, performance comparison of proposed and existing works in terms of number of support data found is displayed. This analysis concluded that the proposed method LPT-FPMDS could find more support data than the previous work TFFPM, and existing work SWTTF. The numerical analysis proves that the proposed LPT-FPMDS shows 2.5% better performance than the previous TFFPM and 5.27% better performance than the existing TFFPM. The time comparison analysis is presented in table 2.

Table 2. Time comparison

| Minimum Support | Time in sec | | |
|---|---|---|---|
| | SWTTF | TFFPM | LPT-FPMDS |
| 0.1 | 5.3840 | 6.2434 | 6.251 |
| 0.2 | 4.2257 | 5.0312 | 5.0456 |
| 0.3 | 3.7458 | 4.3225 | 4.398 |
| 0.4 | 3.9524 | 4.1353 | 4.21 |

| | | | |
|---|---|---|---|
| 0.5 | 3.5433 | 3.7832 | 3.793 |
| 0.6 | 3.6773 | 4.5639 | 4.615 |

Figure 4 represents the comparison of consumed time by proposed and existing techniques.

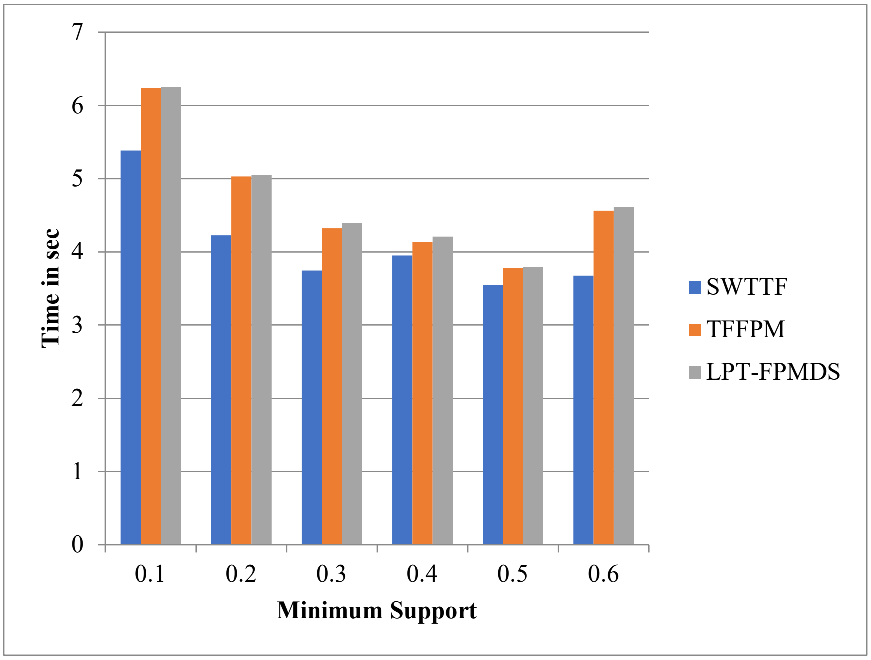


Figure 4. Time consumption comparison analysis

In figure 4, the performance evaluation of proposed and existing models in terms of time consumption is displayed. This analysis concluded that the proposed method LPT-FPMDS could perform with lesser time consumption than the previous work TFFPM and the existing work TFFPM. The numerical analysis proves that the proposed LPT-FPMDS shows 0.83% better performance than the previous work TFFPM and 15.42% better performance than existing TFFPM. The discovered frequent itemsets are given in table 3.

Table 3. Frequent itemsets found comparison

| Minimum Support | Frequent Itemsets found | | |
|---|---|---|---|
| | SWTTF | TFFPM | LPT-FPMDS |
| 0.1 | 436 | 436 | 448 |
| 0.2 | 192 | 192 | 195 |
| 0.3 | 73 | 73 | 85 |
| 0.4 | 52 | 52 | 59 |
| 0.5 | 40 | 40 | 43 |
| 0.6 | 29 | 29 | 36 |

Figure 5 displays the graphical comparison of number of discovered frequent itemsets.

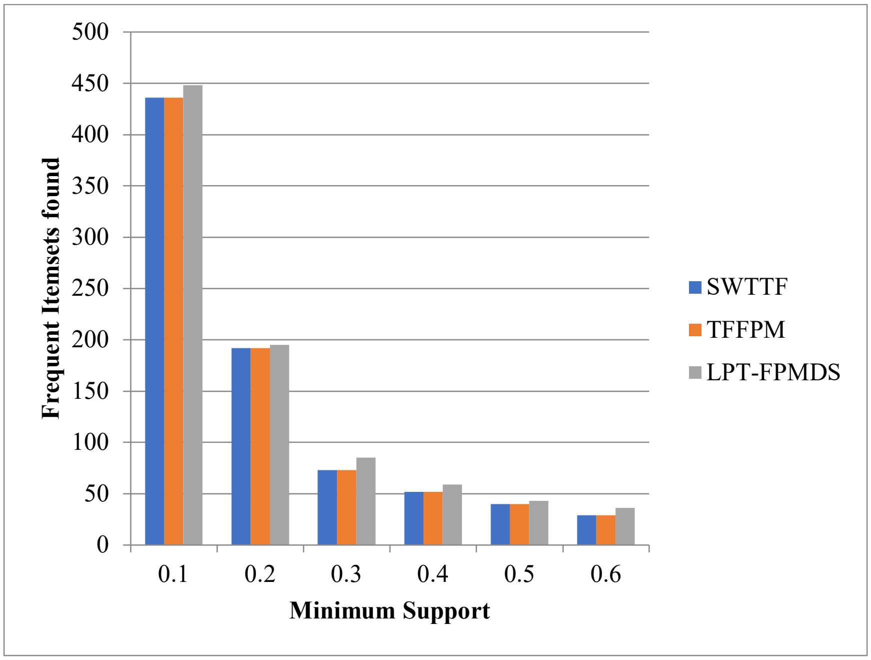


Figure 5. Number of frequent itemset found comparison analysis

Figure 5, shows the comparison analysis of the proposed and existing model in terms of number of frequent itemsets found. This analysis concluded that the proposed method LPT-FPMDS shows 5.35% better performance than TFFPM and SWTTF. The number of rules taken for comparison is portrayed in table 4.

Table 4. Number of rules found comparison

| Minimum Support | Number of rules found | | |
|---|---|---|---|
| | SWTTF | TFFPM | LPT-FPMDS |
| 0.1 | 72 | 82 | 87 |
| 0.2 | 61 | 76 | 79 |
| 0.3 | 58 | 69 | 73 |
| 0.4 | 47 | 67 | 73 |
| 0.5 | 36 | 53 | 55 |
| 0.6 | 33 | 48 | 53 |

The graphical comparison of the number of rules found is given in the following figure 6.

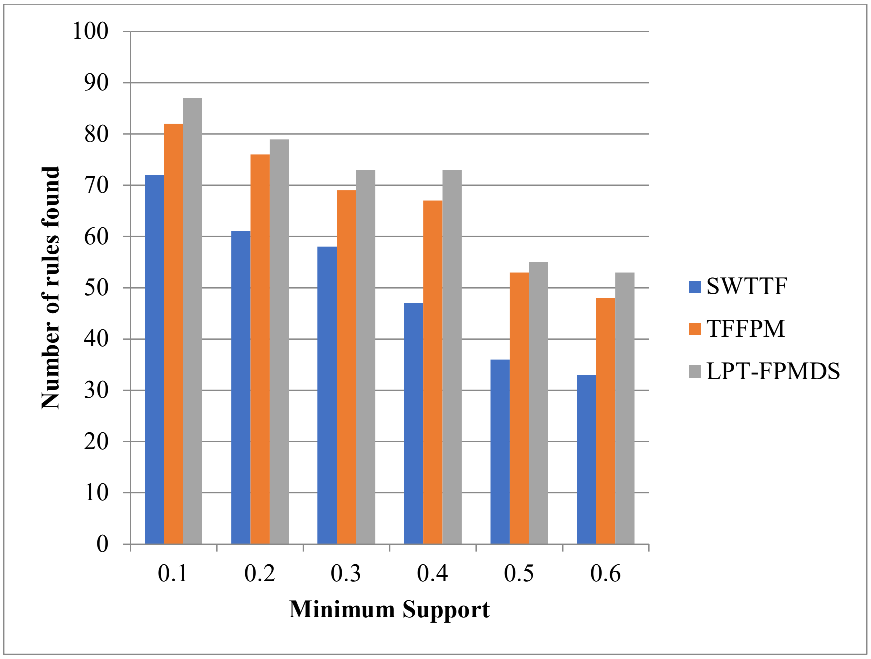


Figure 6. Number of rules found comparison analysis

In figure 6, comparative analysis is performed for proposed and existing models in terms of number of rules found This analysis concluded that the proposed method LPT-FPMDS can perform with lesser number of rules found than the previous work TFFPM and the existing work SWTTF. The numerical analysis proves that the proposed LPT-FPMDS shows 6.33% better performance than the previous work TFFPM and 36.8% better performance than existing TFFPM.

## 5. Conclusion

Optimal attribute selection is done using Hybrid crow swarm Optimization with Cat Swarm Algorithm. Here fitness values considered are accuracy, error rate, Information Gain and Gain Ratio. Based on selected attributes anomalous rule generation is performed. Rule pruning is performed based on Laplace accuracy. In this work rule pruning is combined with classification process. In this work linear prefix tree is introduced for the fraudulent detection. The laplace accuracy of rules will be estimated based on rule pruning when the leaf node is reached. The comparative analysis of this research indicates that the proposed model gained better performance in terms of accuracy and error rate than existing models.

## References

[1] Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

[2] Li, L., Lu, R., Choo, K. K. R., Datta, A., & Shao, J. (2016). Privacy-preserving-outsourced association rule mining on vertically partitioned databases. *IEEE Transactions on Information Forensics and Security*, *11*(8), 1847-1861.

[3] Yuan, X. (2017, March). An improved Apriori algorithm for mining association rules. In *AIP conference proceedings* (Vol. 1820, No. 1, p. 080005). AIP Publishing LLC.

[4] Wang, T., Li, N., & Jha, S. (2018, May). Locally differentially private frequent itemset mining. In *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 127-143). IEEE.

[5] Wu, D., Luo, D., Jensen, C. S., & Huang, J. Z. (2019, April). Efficiently Mining Maximal Diverse Frequent Itemsets. In *International Conference on Database Systems for Advanced Applications* (pp. 191-207). Springer, Cham.

[6] Valiullin, T., Huang, J. Z., Yin, J., & Wu, D. (2019, October). A New Approach for Approximately Mining Frequent Itemsets. In *Data Analytics and Management in Data Intensive Domains: XXI In-ternational Conference DAMDID/RCDL'2019 (October 15–18, 2019, Kazan, Russia): Conference Proceedings. Edited by Alexander Elizarov, Boris Novikov, Sergey Stupnikov.–Kazan: Kazan Federal University, 2019.–496 p.* (p. 63).

[7] Ryang, H., & Yun, U. (2017). Indexed list-based high utility pattern mining with utility upper-bound reduction and pattern combination techniques. *Knowledge and Information Systems*, *51*(2), 627-659.

[8] Saha, A., & Khairnar, V. D. (2018, April). Mining Utility Patterns. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 728-731). IEEE.

[9] Gan, W., Lin, J. C. W., Fournier-Viger, P., Chao, H. C., Hong, T. P., & Fujita, H. (2018). A survey of incremental high-utility itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(2), e1242.

[10] Nguyen, T. D., Vu, Q. B., & Nguyen, L. T. (2019, December). Efficient algorithms for mining maximal high-utility itemsets. In *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)* (pp. 428-433). IEEE.

[11] Lin, J. C. W., Ren, S., & Fournier-Viger, P. (2018). MEMU: more efficient algorithm to mine high average-utility patterns with multiple minimum average-utility thresholds. *IEEE Access*, *6*, 7593-7609.

[12] Wang, Z. (2016, May). Mechanism Study for E-commerce Affecting China's Future Marketing Mode. In *2016 International Conference on Economy, Management and Education Technology*. Atlantis Press.

[13] Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, *75*, 38-48.

[14] Jha, S., Guillen, M., & Westland, J. C. (2012). Employing transaction aggregation strategy to detect credit card fraud. *Expert systems with applications*, *39*(16), 12650-12657.

[15] Leu, F. Y., Huang, Y. L., & Wang, S. M. (2015). A Secure M-Commerce System based on credit card transaction. *Electronic Commerce Research and Applications*, *14*(5), 351-360.

[16] Zhang, X., Han, Y., Xu, W., & Wang, Q. (2019). HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*.

[17] Gianini, G., Fossi, L. G., Mio, C., Caelen, O., Brunie, L., & Damiani, E. (2020). Managing a pool of rules for credit card fraud detection by a Game Theory based approach. *Future Generation Computer Systems*, *102*, 549-561.

[18] Lucas, Y., Portier, P. E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., & Calabretto, S. (2020). Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. *Future Generation Computer Systems*, *102*, 393-402.

[19] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, *100*, 234-245.

[20] Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, *51*, 134-142.

[21] de Sá, A. G., Pereira, A. C., & Pappa, G. L. (2018). A customized classification algorithm for credit card fraud detection. *Engineering Applications of Artificial Intelligence*, *72*, 21-29.