CREDIT CARD FRAUD DETECTION USING FREQUENT PATTERN MINING USING FP-MODIFIED TREE AND APRIORI GROWTH

VipinkumarChoudhary[1], Er. DeepikaArora[2]
(1*. Research Scholar, 2*. Asst. Prof.)
RPIIT Bastara, Karnal.

ABSTRACT:-The last few years have seen a major increase in the number of credit card users. This has brought into limelight the need to study users' access patterns to bring to notice the usage patterns of fraudulent users. Data mining is one such field which is aimed at improving the user's access experience and make it more secure. The last few years have seen many different techniques commonly referred to as "Association mining" by in the field of research. The previous works in this field have mostly been concentrated at Apriori and FP growth algorithms or their variants. We considered here two of the variants of Apriori and FP Growth algorithm which are Apriori growth and FP Split tree respectively. We take these as our base algorithms and create a hybrid of these two algorithms which combines the positive effects of each to bring forth an algorithm that has better performance than both the parents. The Apriori growth algorithm has a very highly time complex method for candidate generation as it involves multiple iteration of the database. But the method of generating frequent sets in Apriori growth is effective. On the other hand, the FP Split algorithm further improves the candidate generation method of FP Split tree by avoiding the recursive Subtree generation involved in FP Tree algorithm, but the method to generate frequent sets is very much the same as FP Tree algorithm. The previous works in this field have mostly been concentrated at Apriori and FP growth algorithms or their variants. The experimental results obtained by applying the algorithm on access logs from UCSD DataMining Contest 2009 Dataset (anonymous and imbalanced), will confirm the effectiveness of our algorithm in comparison to the two contemporaries. The results achieved by implementing the proposed technique in Java language confirmed the effectiveness of this methodology. The results were way better than its predecessor FP Tree algorithm and Apriori technique. The algorithms were implemented for various support levels.

I. INTRODUCTION

Every year billions of Euros are lost world wide due to credit card fraud, thus, forcing financial institutions to continuously improve their fraud detection systems. In recent years, several studies have proposed the use of machine learning and data mining techniques to address this problem. However, most studies used some sort of mis-classification measure to evaluate the different solutions, and do not take into account the actual financial costs associated with the fraud detection process. Moreover, when constructing a credit card fraud detection model, it is very important how to extract the right features from the transactional data. This is usually done by aggregating the transactions in order to observe the spending behavioral patterns of the customers.

However, fraud is becoming increasingly more complex and financial institutions are under increasing regulatory and compliance pressures. In order to combat these frauds, banks need more sophisticated techniques of fraud detection.The major problem for e-commerce business today is that fraudulent transactions appear more and more like legitimate ones [1] and simple pattern matching techniques are not efficient to detect fraud. Moreover, the credit card transaction datasets are highly imbalanced and the legal and fraud transactions vary at least hundred times [2]. Generally, in real case, 99% of the transactions are legal while only 1% of them are fraud [3].

II. PROBLEM DEFINED

Frequent Pattern mining here refers to the use the access logs from a credit card server to study the access pattern of fraudulent clients. This research discusses different techniques for content mining on these logs. This research is aimed at detecting fraudulent credit card transactions. We propose to do this by mining the frequent patterns of fraudulent transactions. The base research and the previous works in this field use the following two algorithms :

Apriori algorithm, in spite of being simple, suffers from certain limitations. It is costly to handle a huge number of candidate sets in Apriori algorithm. So we here aim to remove this drawback. And we do it by using FP Tree algorithm instead of Apriori algorithm to generate candidate sets. The FP Tree algorithm scans the database only once thereby reducing the amount of time involved in repeated scans.

The FP Tree mining method might be very efficient and most sought after for use with aFP Tree. But it has a major drawback in its methodology. First the method is very tough to understand . Second it recursively creates multiple FP Trees for mining. This is both time consuming and cumbersome. So we finally use Apriori growth algorithm for mining the FP modified Tree created in step 1. The disadvantage of FP-Growth is that it needs to work out

conditional pattern bases and build conditional FP-tree recursively. It performs badly in data sets of long patterns.
So we proposed here a hybrid technique for frequent pattern mining on credit card access logs using FP Split Tree and Apriori Growth algorithm.

III. PROPOSED METHODOLOGY

The proposed algorithm which is a hybrid of a modified FP Tree creation algorithm and Apriori Growth mining algorithm, is given below in two phases.

The first phase constructs the FP Split Tree which is more efficient way to create candidate sets than FP Tree since the latter involves two complete scans of the database while the former does it once. This impacts the efficiency almost 2 times better. More over the FP Split Tree created by our algorithm involves lesser use of pointers as we don't link each node in the Tree to its predecessor and successor. Rather we maintain a header list separately which maintains a list separately for each of the pages which points to the occurrence of these items in the final tree created.

The second phase involves mining the FP Split tree created using the apriori growth algorithm. This algorithm is more efficient than FP Growth as it does not involve recreating the FP Split trees repeatedly every time in recursion as in FP Growth algorithm thereby reducing the time involved.
Credit card logs mining is one of the most significant fields in the area of data mining. Web Mining is the application of data mining techniques to Web data. A Web log file is a simple text file which keeps record of the requests that are submitted by the user to the server while accessing a website. Key fields contained in log file are: username, IP address, user agent, date, time, number of bytes transferred etc. There have been a large number of data mining algorithms rooted in these fields to perform different data analysis tasks. The top algorithms identified by the IEEE International Conference on Data Mining (ICDM) presented here are among the most influential algorithms for classification, clustering, statistical learning, association analysis, and link mining. We discuss here the two best methods used in this field- Apriori algorithm and FP Tree Algorithm. We propose a hybrid of these two algorithms to propose a solution which combines the +ves of these two algos and removes the –ves.

DATASET USED

amount,hour1,state1,zip1,field1,domain1,field2,hour2,flag1,total,field3,field4,field5,indicator1,indicator2,flag2,flag3,flag4,flag5
38.85,8,FL,342,4,AFIGECHUD.COM,1,8,0,38.85,1688,8,0,0,0,0,1,0,1
38.85,9,GA,300,4,FXXRJTGPDTAOLKVEG.COM,0,9,0,38.85,1174,10,4,0,0,0,1,0,1
12.95,10,CA,939,4,IX.NETCOM.COM,0,10,1,12.95,-754,10,0,0,0,1,1,0,1
12.95,13,VA,223,4,HOTMAIL.COM,1,13,0,12.95,3087,7,0,0,0,1,0,0,1
12.95,14,CA,917,4,COMCAST.NET,0,14,0,12.95,1802,7,0,0,0,0,1,0,2
12.95,14,VA,201,4,HOTMAIL.COM,1,14,0,12.95,-2724,6,0,1,0,1,0,0,1

ALGORITHM

The proposed algorithm which is a hybrid of a modified FP SplitTree creation algorithm and Apriori Growth mining algorithm, is given below in two phases.
The first phase constructs the FP Split Tree which is more efficient way to create candidate sets than FP Tree since the latter involves two complete scans of the database while the former does it once. This impacts the efficiency almost 2 times better. More over the FP Split Tree created by our algorithm involves lesser use of pointers as we don't link each node in the Tree to its predecessor and successor. Rather we maintain a header list separately which maintains a list separately for each of the pages which points to the occurrence of these items in the final tree created.
The second phase involves mining the FP Split tree created using the apriori growth algorithm. This algorithm is more efficient than FP Growth as it does not involve recreating the FP Split trees repeatedly every time in recursion as in FP Growth algorithm thereby reducing the time involved.

<table>
<tr><td>Content</td><td>Count</td><td rowspan="2">Link_Sibling</td></tr>
<tr><td colspan="2">List</td></tr>
<tr><td colspan="3">Link_Child</td></tr>
</table>

Node Structure for FP Tree

Phase 1.
FP Modified Tree construction
Step-1. Scanning the database to create equivalence class of item. Let the equivalence class of item be EC,= {Tid I Tg are the identifier of transaction ti; 1 is an item of ti).
Step-2. Calculating support to filter out non-frequent items. The support of each item I- refers to the number of records contained in the equivalence class EC,.LetpCLl denote the support of the equivalence class EC,. After calculating the supports of items, we delete the items whose supports are below the predefined minimum support
Step-3. After generating frequent items, the equivalence class ofitem is then converted into nodes for the construction ofFP-split tree. To facility tree traversal, a header table is built inadvanced so that each item can point to its first occurrence inthe FP-split tree.
Step-4. While constructing the FP-split tree, a root will be generated, which is a dummy node.
Phase II.
Apriori Growth algorithm for mining.

Name
*Tablelink
Count

Data Structure of Node of Header Table

Name
*TableLink
*parent
*Child
Count

Algorithm

Input : pagelist2 : list of pages with equivalence class satisfying support count

Tree : Nodes of tree as a list.

Sup : minimum support

Output :frequent item sets 'data'

```
1.      Repeat following step while scanning pagelist2 till
end
            • Create list containing single item from
              pagelist2
            • Add this list to mylist1

        [End of repeat]
[mylist1 is 1 frequent itemset]
2.              Add mylist1 to data
3               Repeat for k=2,3,4,….
4.              $C_k$=getCandidate(k)
5.              If [$C_k$]=0 then
                Goto step 6
                Else
                Add $C_k$ to data
                End if
                [End of Repeat]
6.              End
```

Algo :getCandidate

```
1.              List1=k-1 frequent item dataset from Data
2.              N=size(list1)
                Initialize mylist as a blank list to contain
generated frequent item dataset
3.              Repeat for I=1 to n-1
4.              Repeat for j=I+1 to n
5               l1=list1[I]
6               l2=list1[j]
7               Remove the last elements from l1 and l2
8               if l2 is a subset of l1 then
                Flist=append last element of l2 at end of l1
                [end of while]
Step 9.         If count(flist)>=supp then
Add flist to mylist
Else
return null
Step 10. end
```

IV. RESULTS

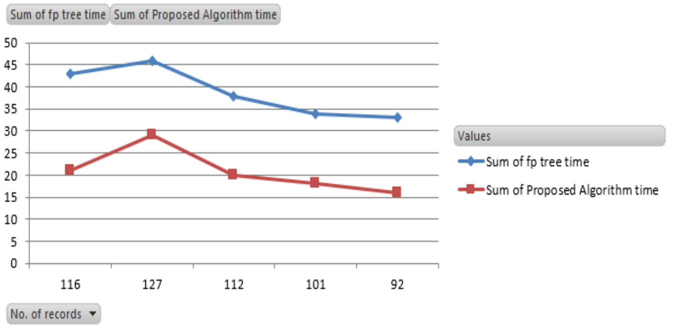

Figure 4

No. of records	FP tree time	Proposed Algorithm Time
127	46	29
116	43	21
112	38	20
101	34	18
92	33	16

The above graph shows the comparison between FP Tree and FP Split tree methods for support count 3. We can see that the time taken by proposed algorithm is always better than the traditional method. The difference would be more clearly visible if we get logs of few days time for some organization where the incoming requests are high in number. The proposed algorithm has been tested on logs received from data-mining contest 2009 task1. The more is the no. of records, the better visible is the difference between the efficiency of the two algorithms.

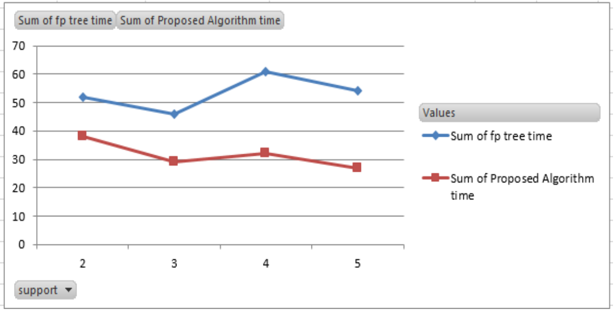

Figure 5

Support	FP tree time	Proposed Algorithm Time
2	52	38
3	46	29
4	61	32
5	54	27

The above graph shows the comparison between FP Tree and FP Split tree methods for different support counts for a fixed no. of records.

Frequent sets generated for threshold value 4:-

Mining the FP split tree created

Added 1 frequent itemset

[[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106], [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [122], [123], [124], [125], [126], [127], [128], [129], [130], [131], [132], [133], [134], [135], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146], [147], [148], [149], [150], [151], [152], [153], [154], [155], [156], [157], [158], [159], [160], [161], [162], [163], [164], [165], [166], [167], [168], [169], [170], [171], [172], [173], [174], [175], [176], [177], [178], [179], [180], [181], [182], [183], [184], [185], [186], [187], [188], [189], [190], [191], [192], [193], [194], [195], [196], [197], [198], [199], [200], [201], [202]]

Added 2 frequent itemset

[[0, 5], [0, 51], [0, 60], [0, 71], [0, 88], [0, 110], [0, 113], [0, 174], [5, 71], [6, 148], [60, 110], [92, 134], [93, 160], [116, 141]]

Added 3 frequent itemset

[[0, 5, 71], [0, 60, 110]]

The above results clearly demonstrate that the dataset taken into consideration when worked for threshold value 4, produce the above shown frequent patterns. So we can study frequent patterns for fraudulent records and come to know the user access patterns for these activities. These patterns can be now matched with the next upcoming users access patterns and any successful match should enable the detection of fraudulent user.

V. CONCLUSION

Data mining is one such field which is aimed at improving the users access experience and make it more secure. The previous works in this field have mostly been concentrated at Apriori and FP growth algorithms or their variants. We considered here two of the variants of Apriori and FP Growth algorithm which are Apriori growth and FP Split tree respectively. We take these as our base algorithms and create a hybrid of these two algorithms which combines the positive effects of each to bring forth an algorithm that has better performance than both the parents. The experimental results obtained by applying the algorithm on access logs from UCSD DataMining Contest 2009 Dataset (anonymous and imbalanced), will confirm the effectiveness of our algorithm in comparison to the two contemporaries. The results achieved by implementing the proposed technique in Java language confirmed the effectiveness of this methodology. The results were way better than its predecessor FP Tree algorithm and Apriori technique. The algorithms were implemented for various support levels.

This hybrid of Apriori growth and FP Split algorithms can be used in other researches involving association mining like social media mining,wsn data mining etc. Another task can be to further try and improve the complexity of this proposed technique to evolve a better method.

ACKNOWLEDGEMENT

I owe my thanks to the almighty first and then to my guide Er. Ms. Deepika Arora who helped me to accomplish this research work successfully. I am also grateful to my family and friends for all their kind and unconditional support.

REFERENCES

[1] Alejandro Correa Bahnsen∗, DjamilaAouada, AleksandarStojanovic, BjörnOttersten, “Feature engineering strategies for credit card fraud detection”, Expert Systems With Applications, Elsevier 2016.

[2] Tanmay Kumar Behera and SuvasiniPanigrahi, “Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering & Neural Network”, Second International Conference on Advances in Computing and Communication Engineering, IEEE 2015.

[3] NedaSoltaniHalvaiee and Mohammad KazemAkbari, “A novel model for credit card fraud detection usingArtificial Immune System”, Applied Soft Computing 24 Elsevier (2014) 40–49.

[4] Andrea Dal Pozzolo, Olivier Caelen, Yann-A El Le Borgne, Serge Waterschoot And GianlucaBontempi, “Learned lessons in credit card fraud detection from a practitioner perspective”, Expert Systems with Applications 2015.

[5] K. R. Seeja and MasoumehZareapoor,” FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent ItemsetMining⌋e Scientific World Journal Volume 2014, Article ID 252797.

[6] Priyanka S. Panchal, Prof. Urmi D. Agravat, “Hybrid Technique for User’s Web Page Access Prediction based on Markov Model”, 4th ICCCNT 2013 ‖ July 4-6, 2013 ‖ Tiruchengode, India

[7] Harendra Singh, Ashish Kumar Srivastava, SitendraTamrakar, “A Modified FP-Tree Algorithm for Generating Frequent Access Patterns”, JECET ‖ June – August-2013; Vol.2.No.3 ‖ 730-740 ‖ E-ISSN: 2278–179X

[8] Omer Adel Nasser & Dr. Nedhal A.AL Saiyd, The Integerating Between Web Usage Mining and Data Mining Techniques,5th Conference on CSIT,IEEE(2013)

[9] ShipraKhare, Prof Vivek Jain, Prof ManojRamaiya, Implementation of Web Usage Mining with Customized Web Log Using FP Growth Algorithms, International

Journal of Engineering & Managerial Innovations (IJEMI) ‖ ISSN: 2321-693X ‖ Volume I (II), September (2013)

[10] Jun Yang, Z. Li, Wei Xiang, An Improved Apriori Algorithm Based on Features, IEEE (2013)

[11] BinaKotiyal, Ankit Kumar, Bhaskar Pant, R.H. Goudar, ShivaliChauhan and SonamJunee, "User Behavior Analysis in Web Log through Comparative Study of Eclat and Apriori", Proceedings of 7th International Conference on Intelligent Systems and Control (ISCO 2013) ‖ 978-1-4673-4603-0

[12] MajaDimitrijevic, TanjaKrunic, "Association rules for improving website effectiveness: case analysis", Online Journal of Applied Knowledge Management ‖ Volume 1, Issue 2, 2013

[13] Kirti S. Patil, Sandip S. Patil, "Sequential Pattern Mining Using Apriori Algorithm & Frequent Pattern Tree Algorithm", IOSR Journal of Engineering (IOSRJEN) ‖ e-994 ISSN: 2250-3021, p-ISSN: 2278-8719 ‖ Vol. 3, Issue 1 (Jan. 2013), ‖V4‖ PP 26-30

[14] YunLong Song and RamWei, Research on Application of Data Mining based on FP Growth Algorithm for Digital Library, IEEE (2011)

[15] Sandeep Singh Rawat, Lakshmi Rajamani, "Discovering Potential User Browsing Behaviors Using Custom-Built Apriori Algorithm", International Journal of Computer Science & Information Technology (IJCSIT) ‖ Vol.2, No.4, August 2010

[16] Goswami D.N., ChaturvediAnshu, Raghuvanshi C.S., "An Algorithm for Frequent Pattern Mining Based On Apriori", (IJCSE) International Journal on Computer Science and Engineering ‖ Vol. 02, No. 04, 2010 ‖ 942-947

[17] RakeshAgrawal, RamakrishnanSrikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB Conference, Santiago, Chile, 1