International Journal of Electrical Engineering and Technology (IJEET)
Volume 11, Issue 10, December 2020, pp. 440-448, Article ID: IJEET_11_10_055
Available online at https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=10
ISSN Print: 0976-6545 and ISSN Online: 0976-6553
DOI: https://doi.org/10.34218/IJEET.11.10.2020.055

© **IAEME** Publication **Scopus** Indexed

A COMPREHENSIVE ANALYSIS OF FREQUENT ITEMSETS MINING KEY ALGORITHMS USING DIVERSE DATASETS

Dr. Ramah Sivakumar
Assistant Professor, Department of Computer Science
Bishop Heber College (Affiliated to Bharathidasan University),
Tiruchirappalli, Tamil Nadu, India

ABSTRACT

In Frequent Itemset mining (FIM), the main process includes looking for combination of itemsets from the data sets. The search process can be done using a variety of mining algorithms such as Apriori, the FPGrowth, and eclat algorithms which are some of the implementations of frequent itemsets search methods. By analyzing the mined data, proper decision making may be done which has huge benefits such as increased revenue, cost cutting, improved competitive advantages and so on. When the size of the dataset is large, the mining process requires more time, and also due to a heavy computation by the algorithm it involves significant memory consumption to mine. Efficient algorithms are required to mine the hidden patterns of the frequent itemsets within a shorter processing time and also with less memory consumption while the volume of data increases. In this research work different real-time datasets whose characteristics are completely different, are used with the aim of knowing the behavior and influence of the datasets on the algorithms. In this paper, an analysis and a comparison of key FIM algorithms is done in order that more efficient FIM algorithms are often developed.

Key words: Data mining, Frequent Itemset Mining (FIM), Frequent Pattern Mining (FPM).

Cite this Article: Ramah Sivakumar, A Comprehensive Analysis of Frequent Itemsets Mining Key Algorithms using Diverse Datasets, *International Journal of Electrical Engineering and Technology (IJEET)*. 11(10), 2020, pp. 440-448.
https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=10

1. INTRODUCTION

Frequent Itemset mining (FIM) uses mathematical calculations, and statistical techniques to extract and identify useful information and related knowledge from various large databases. Initially, Frequent Itemset Mining was introduced as a method for market basket analysis[1]. The main aim of it is to find the customers buying behavior in supermarkets. In specific, to find out the sets of products those are frequently bought together. It is often found patterns are transformed into association rules, for example, if a customer buys bread and butter, then she/he will also probably buy cheese or jam. The analysis paved way to improve arrangement of products in shelves, on a catalog's pages and also supports cross-selling and suggestion of other products, product bundling which improves the overall business profit. Presently, the application of FIM also includes Fraud detection, technical dependence analysis, and fault localization, players' behavior, server performance, system functionality and so on. Many researchers [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] done an analysis of the transactional databases in the recent years.

1.1. Background

The basic terminologies of frequent Itemset mining (FIM) are:

Table 1 Terminologies

Items	$I = \{i_1, \ldots, i_n\}$
Itemset, transaction	$P, T, \subseteq I$
Transactional dataset	$D = \{T_1, \ldots, T_m\}$

Definitions:

Given a set of items $I = \{i_1, i_2, \ldots, i_n\}$, a transactional dataset $D = \{T1, \ldots, Tm\}$, and a minimum support θ. The need is the set of itemset P that is freq (P) $\geq \theta$.

2. DATASET REPRESENTATIONS

The dataset can be represented horizontally, vertically as well as in matrix formats.

Dataset H_D in Horizontal Representation:

Table 2 horizontal representation

Transactions	Items
1	a,d,c
2	b,c,d
3	a,c,e
4	a,c,d,e
5	a,e
6	a,c,d
7	b,c
8	a,c,d,e
9	b,c,e
10	a,d,e

Dataset V_D in Vertical Representation:

Table 3 Vertical representation

Items	a	b	c	d	e
Transactions	1	2	2	1	1
	3	7	3	2	3
	4	9	4	4	4
	5		6	6	5
	6		7	8	8
	8		8	10	9
	10		9		10

Dataset M_D in Matrix Representation:

Table 4 Matrix representation

	a	**b**	**c**	**d**	**e**
1	1	0	0	1	1
2	0	1	1	1	0
3	1	0	1	0	1
4	1	0	1	1	1
5	1	0	0	0	1
6	1	0	1	1	0
7	0	1	1	0	0
8	1	0	1	1	1
9	0	1	1	0	1
10	1	0	0	1	1

The Anti-monotonicity property:

Given a transaction database D over items I and two itemsets X, Y:

$X \subseteq Y \Rightarrow freq(Y) \leq freq(X)$-----→ 1

Generation of Frequent Itemsets

After the first scan of the database frequent-1 itemsets could be found out by the counts for each item, and the items that meet the minimum support are collected. This process is referred to as Level-1 process and level by level frequent itemsets are found. Frequent itemsets found at the first level are termed as L1. Then L1 is used to find L2, and then L2 is used to find L3, and so on, until no more frequent items could be found out. Until there is no more suitable combinations of items could be found, scanning the database is done for each Lk.

3. RELATED RESEARCH

Using the join and prune technique Agrawal et.al proposed the classic algorithm Apriori [1]. The base of Apriori algorithm is candidate generation and pruning. Based on the given support count more number of candidates are generated and then they are pruned. This includes repeated scanning of databases according to its data volume and excessive I/O operations also, which in turn leads to more time and space complexity. The above reason is the main disadvantage of Apriori algorithm.

Another algorithm without candidate generation method is FPGrowth which involves tree concept[2]. The tree is constructed using the header table technique. The recursive conditional pattern base generation and sub conditional pattern trees are the main drawbacks in FPGrowth.

By the process of matrix generation and using array, maximal frequent pattern mining algorithm was devised by Peng Hui-ling at al.[9] which works on one time database scanning. By the frequency of itemsets, the itemsets are derived in descending order. The itemsets' frequency list is used to generate the FP- tree which represents the whole database. Then the conditional FP-tree replaces the whole database which is used to generate frequent patterns.

Apriori and FPGrowth algorithms use horizontal database formats while in the later phase an algorithm called Eclat[5] was proposed which use vertical database format for mining process.

To get the best out of these algorithms optimizations were made in the later periods and enhanced algorithms were also proposed by the developers, which were based on the above said algorithms. Nodes and header tables are the base for tree based algorithms. For optimization in tree based algorithms different types of data structures are also used. Bay Vo, et.al.[20] proposed with N-list technique, and Zhi-Hong Deng[21] proposed DiffNodeset, a novel and more efficient itemset representation, for mining frequent itemsets. Based on the DiffNodeset structure, an efficient algorithm, named dFIN is presented, for mining frequent itemsets. To achieve high efficiency, dFIN finds frequent itemsets using a set-enumeration tree with a hybrid search strategy and directly enumerates frequent itemsets without candidate generation under some case. DP-Apriori algorithm which used transaction splitting was proposed by Xiang Cheng et al.,[22] with a support estimation technique to prevent information loss. Ling Chen et.al., proposed MSPM algorithm for patterns in multiple biological sequences. This approach used pattern extending technique based on prefix tree and mines frequent patterns without candidate generation**.**

4. STUDY OF SOME FREQUENT ITEMSET MINING ALGORITHMS

Apriori Algorithm

The first classical algorithm proposed by Agrawal at al. [1] was Apriori. Apriori Algorithm is one of the classic Itemset mining algorithms that is used to find frequent itemsets from the datasets. The parameter needed to find frequent itemsets is: *minimum support*. Apriori uses the breadth-first-search technique to find frequent itemsets combinations.

By searching repeatedly all the frequent itemsets from all the items' combinations in the dataset is found out using candidate generation Ck-1 process. This looping process occurs while searching and selecting combinations in the lattice tree for pruning process and to determine the appropriate combinations. This is referred to as the join and prune technique. Many researchers had optimized Apriori algorithm for larger and more complex transaction datasets in terms of memory usage and processing time.

Pitfalls in Apriori

- Generation of candidate itemsets. If the itemsets in the database is enormous, candidate itemsets may be large in number.
- Not cost effective as multiple scans of the database are done to check the support of each itemset generated.

FP-Growth Algorithm

Without candidate generation and using tree data structure or (FP-Tree), this algorithm is considered as improvement to Apriori. Han at al. [2] initially worked on Frequent pattern growth algorithm. The database is represented in the form of a tree called a frequent pattern tree or FP tree in FP growth algorithm. FP-tree is a compressed data storage structure. The root node of the FP-tree holds null whereas the lower nodes hold the itemsets, and each node represents an item of the itemset.

Through mapping in the corresponding paths the FP-tree is built. If the transactions have the same item in the same path then it is overwritten. The more the re-occurrences of the same item, then the more effective process of compression is carried out.

The FP-Growth method is composed of three stages namely:

- Conditional pattern base generation
- FP-tree conditionals generation
- Searching frequent itemsets

Through these stages frequent Itemsets are mined successfully.

ECLAT Algorithm

ECLAT algorithm[5] uses vertical data format for mining frequent itemsets, whereas Apriori and FP growth algorithms mine frequent itemsets using horizontal data format. The data in the horizontal data format is transformed into vertical format. 2-itemsets, 3 itemsets, upto k itemsets are formed using vertical data format until no candidate itemsets are found. As the transaction set carries the count of occurrence of each item in the transaction this method k+1 itemsets are formed without scanning the database. When the number of transactions is more that is when the dataset is big, memory consumption and processing time for intersecting the sets is also more and that leads to bottleneck.

5. RESEARCH FRAMEWORK

This research work intends to find out the behavior of the key Itemset mining algorithms against various datasets of different characteristics, and to analyze and compare the algorithms. The frequent itemset search was conducted and experiments were valued with minimum-support values on datasets ranging from 0.01 - 0.05. Average values are calculated for final consideration. Memory usage was calculated in megabytes (MB) and the processing time in milliseconds (MS). The scalability of the algorithms was also calculated with the added transactions.

Datasets & Tool

The datasets used in this research are downloaded from the FIMI Dataset Repository published by IBM Almaden Quest Research Group sourced from http://fimi.ua.ac.be/data/ .

For this research work, SPMF [14] which is an open-source software and data mining library written in Java, specialized in pattern mining (the discovery of patterns in data) is used. This tool mines the data patterns using itemset mining algorithms. Machine with configuration of windows 10 operating system and 4GB of RAM is used.

Dataset Characteristics

Table 5 Dataset Characteristics

Datasets	Transactions	Distinct Items	Avg. Transaction Size	Area	File size(KB)
Chess	3196	36	37	Game	338
Mushroom	8416	22	23	Life	598
Pumsb	49046	7116	74	Census	4621
Accident	340183	468	22	Traffic	9216

6. RESULTS AND DISCUSSION

Processing Time (MS)

Table 6 Processing Time (MS)

Dataset/ Algorithm	FPGrowth	Apriori	Eclat
Pumsb	1156	18866	5380
Mushroom	1560	24970	17340
Chess	5625	96802	43899
Accident	7250	113891	52843

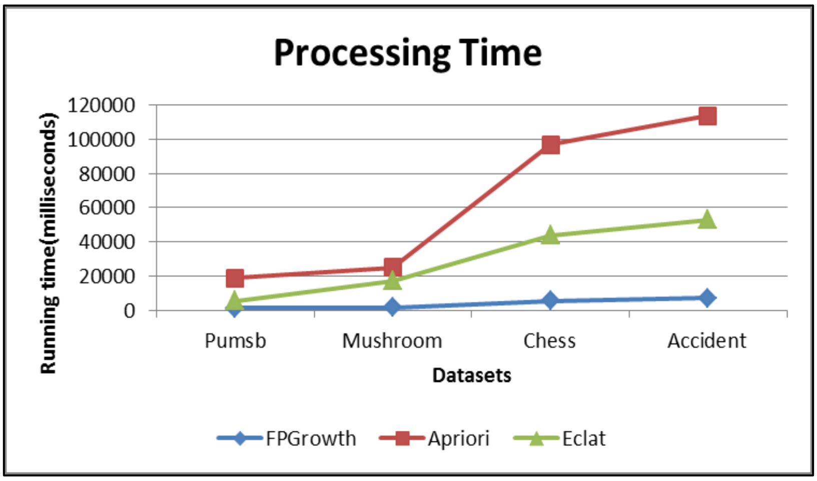

Figure 1 Processing Time (MS)

Memory Consumption (MB)

Table 7 Memory Consumption (MB)

Dataset/Algorithm	FPGrowth	Apriori	Eclat
Pumsb	85.53	208.98	387.23
Mushroom	283.88	351.97	480.08
Chess	431.08	502.58	631.08
Accident	593.84	621.78	893.52

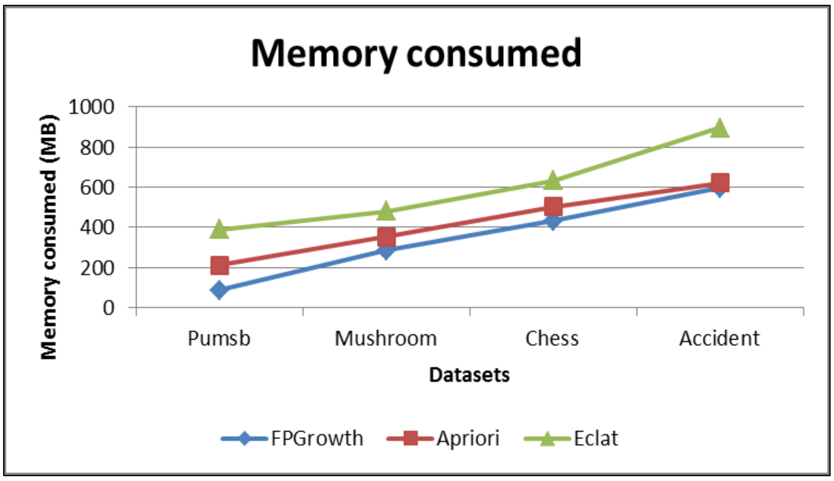

Figure 2 Memory Consumption (MB)

It is observed that the itemsets are mined in less time and with less space utilization while using Pumsb dataset. Though the average transaction size of accident dataset is less compared to others it takes more processing time and occupies more space for mining process as the total number of transactions is more compared to other datasets. The size of the mushroom dataset is less compared to other datasets. Still the processing time and memory consumption of algorithms is more in mushroom compared to pumsb dataset as it is a denser one. The processing time and memory consumption of algorithms for Chess dataset more compared with pumsb and mushroom datasets as it is more denser than others.

Scalability - Processing Time(ms)

Table 8 Scalability - Processing Time(ms)

Algorithms	Added Transactions		
	100	300	500
Processing time			
FPGrowth	7.85	10.07	13.71
Eclat	11.3891	14.80583	19.247579
Apriori	15.2843	19.86959	25.830467

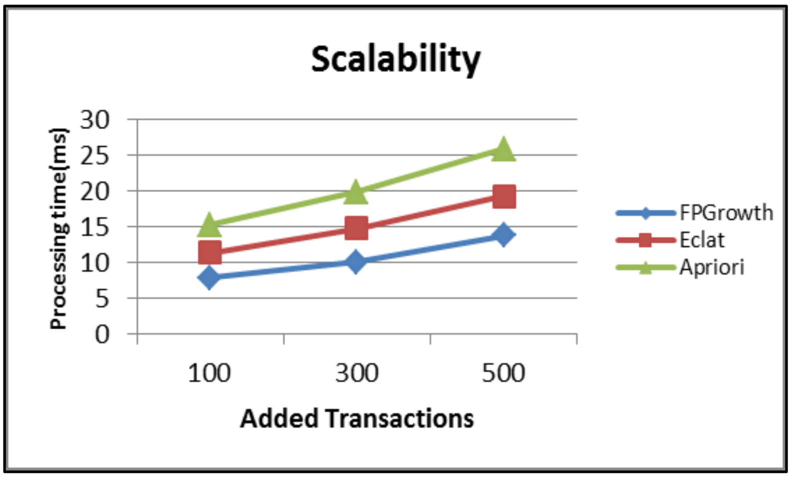

Figure 3 Scalability - Processing Time(ms)

The scalability experiment was performed with the pumsb dataset. This dataset was considered as it took less processing time compared with the other datasets. The experiment was conducted by adding upto 500 transactions and the minsup parameter was set to the average level. The main goal is to find out how far the added transactions influence the execution of the algorithms. Results are shown above. It could be seen that FPGrowth is more scalable than the other algorithms.

7. CONCLUSION

It is still an open question that certain algorithms accomplish very well or very poor on some datasets. In this setting, a thorough experimental analysis of datasets with respect to frequent itemsets is conducted. The distribution of frequent itemsets with respect to itemsets size is also analyzed. The outcome of these tests is classifying the datasets with respect to minsup variations and robust to find out the efficiency of the algorithms. The research work is carried out to examine the performance of various key Itemset mining algorithms when applied to real time datasets with different characteristics. The results of the analysis show that the performance of the algorithms varies according to the dataset in terms of processing time and memory usage. The scalability of the algorithms are also analyzed by adding transactions to the datasets and examined with their processing time. It is found out that there is still need to propose adaptive algorithms with respect to the characteristics of the datasets to mine all the frequent itemsets in single database scan. It should also be scalable and support the mining process without the need to rescan the whole database during addition or deletion of transactions in the database that is the execution strategy should change dynamically during run time in accordance with the dataset.

REFERENCES

[1] Agrawal, R., Imielminski, T., Swami, A.: “Mining Association Rules Between Sets of Items in Large Databases”. In: Proc. ACM Intern. Conf. on Management of Data, pp. 207-216, ACM Press (1993)

[2] Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation. Data Mining and Knowledge Discovery 8, 53-87 (2004)

[3] Rakesh Agrawal, Ramakrishnan Srikant, “Fast Algorithms for Mining Association Rules” Proceedings of the 20th VLDB Conference Santiago, Chile, 1994

[4] R. Agarwal, C. C. Aggarwal, and V. V. V. Prasad, “Depth-first Generation of Long Patterns”, *ACM KDD Conference*, 2000. Also available as IBM Research Report, RC21538, July 1999.

[5] M. J. Zaki. Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering, 12(3), pp. 372–390, 2000.

[6] Zaki, M. J., Gouda, K.: Fast vertical mining using diffsets. In: Proc. of the ninth ACM SIGKDD international conference on Knowledge Discovery and Data mining, pp. 326-335. ACM Press (2003)

[7] G. Liu, H. Lu and J. X.Yu. AFOPT:An Efficient Implementation of Pattern Growth Approach, FIMI Workshop, 2003.

[8] Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd ed. Morgan Kaufmann, San Francisco (2006)

[9] Peng Hui-ling, Shu Yun-xing “A new FP-tree-based algorithm MMFI for mining the maximal frequent itemsets”, IEEE International Conference on Computer Science and Automation Engineering (CSAE), May 2012

[10] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent Pattern Mining: Current Status and Future Directions, Data Mining and Knowledge Discovery, 15(1), pp. 55–86, 2007.

[11] Fournier-Viger, P., Wu, C.-W., Tseng, V. S.: Mining Top-K Association Rules. In Proc. of the 25th Canadian Conf. on Artificial Intelligence (AI 2012). LNAI, vol. 7310, pp. 61-73. Springer, Heidelberg (2012)

[12] O.Jamsheela, Raju G., “Frequent itemset mining algorithms: A literature survey”, 2015 IEEE International Advance Computing Conference (IACC), IEEE, 10.1109/IADCC.2015.7154874

[13] Frédéric Flouvat · Fabien De Marchi · Jean-Marc Petit, “A new classification of datasets for frequent itemsets”, © Springer Science + Business Media, LLC 2009

[14] Fournier-Viger, P., Lin, C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H. T. (2016). The SPMF Open-Source Data Mining Library Version 2. Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III, Springer LNCS 9853, pp. 36-40.

[15] Chin-Hoong Chee, Jafreezal Jaafar, Izzatdin Abdul Aziz, Mohd Hilmi Hasan & William Yeoh, “Algorithms for frequent itemset mining: a literature review”, Artificial Intelligence Review, 52, pages 2603–2621 (2019), https://doi.org/10.1007/s10462-018-9629-z

[16] Anshu singla, Parul Gandhi, “A comparative study of frequent itemset mining algorithms”, Journal of critical reviews, ISSN- 2394-5125 VOL 7, ISSUE 19, 2020

[17] Chunkai Zhang, Panbo Tian, Xudong Zhang, Qing Liao, Zoe L. Jiang, Xuan Wang, HashEclat: an efficient frequent itemset algorithm, International Journal of Machine Learning and Cybernetics, © Springer-Verlag GmbH Germany, part of Springer Nature 2019 https://doi.org/10.1007/s13042-018-00918-x

[18] Shashi Raj, Dharavath Ramesh,M. Sreenu, Krishan Kumar Sethi, “EAFIM: efficient apriori-based frequent itemset mining algorithm on Spark for big transactional data”, Knowledge and Information Systems, © Springer-Verlag London Ltd., part of Springer Nature 2020, https://doi.org/10.1007/s10115-020-01464-1,

[19] José María Luna, Philippe Fournier-Viger, Sebastián Ventura, "Frequent itemset mining: A 25 years review", WIREs Data Mining KnowledgeDiscov.2019;e1329,wires.wiley.com/dmkd, © 2019 Wiley Periodicals, Inc

[20] Bay Vo, Tuong Le, Frans Coenen, Tzung-Pei Hong "Mining frequent itemsets using the N-list and subsume concepts", International Journal of Machine Learning and Cybernetics, @ Springer Berlin Heidelberg, 2016.

[21] Zhi-Hong Deng, "DiffNodesets: An efficient structure for fast mining frequent itemsets", Applied Soft Computing, Elsevier, vol.41,pages 214-223, 2016.

[22] Xiang ChengSen SuShengzhi XuShengzhi XuZhengyi Li, "DP-Apriori: A Differentially Private Frequent Itemset Mining Algorithm Based on Transaction Splitting", Computers & Security, February 2015.

[23] Anshu singla1, Parul Gandhi, "A comparative study of frequent itemset mining algorithms", Journal Of Critical Reviews, ISSN- 2394-5125 VOL 7, ISSUE 19, 2020.

[24] Kalaiarasi, K., and R. Gopinath. "Stochastic Lead Time Reduction For Replenishment Python-Based Fuzzy Inventory Order Eoq Model With Machine Learning Support." Technology (IJARET) 11.10 (2020): 1982-1991.

[25] Kalaiarasi, K., and R. Gopinath. " Fuzzy Inventory Eoq Optimization Mathematical Model." International Journal of Electrical Engineering and Technology (IJARET) 11.8 (2020): 169-174.

[26] Priyadharshini, D., R. Gopinath, and T. S. Poornappriya. "A Fuzzy Mcdm Approach For Measuring The Business Impact Of Employee Selection." International Journal of Management (IJM) 11.7 (2020): 1769-1775.

[27] V Upendran, and R. Gopinath, "Feature Selection Based On Multicriteria Decision Making For Intrusion Detection System", International Journal of Electrical Engineering and Technology, 11.5 (2020): 217-226.

[28] R. Gopinath, A. Chitra, R. Kalpana. "Emotional Intelligence And Knowledge Management- A Relationship Study", International Journal of Advanced Research in Engineering and Technology, 11.11 (2020): 2363-2372.

[29] M. Subhashini, and R. Gopinath. "Employee Attrition Prediction In Industry Using Machine Learning Techniques", International Journal of Advanced Research in Engineering and Technology, 11.12 (2020): 3329-3341.

[30] S. Rethinavalli, and R. Gopinath. "Botnet Attack Detection In Internet Of Things Using Optimization Techniques", International Journal of Electrical Engineering and Technology, 11.10 (2020): 412-420.

[31] S. Rethinavalli, and R. Gopinath. "Classification Approach-Based Sybil Node Detection In Mobile Ad Hoc Networks", International Journal of Advanced Research in Engineering and Technology, 11.12 (2020): 3348-3356.

[32] T.S. Poornappriya., and R. Gopinath. "Application Of Machine Learning Techniques For Improving Learning Disabilities", International Journal of Electrical Engineering and Technology, 11.10 (2020): 403-411.

[33] T.S. Poornappriya., and R. Gopinath. "Rice Plant Disease Identification Using Artificial Intelligence Approaches", International Journal of Electrical Engineering and Technology, 11.10 (2020): 392-402.

[34] R. Gopinath., and T.S. Poornappriya. "An Analysis Of Human Resource Development Practices In Small Scale Startups", International Journal of Advanced Research in Engineering and Technology, 11.11 (2020): 2475-2483.