Enhanced Compressed Maximal Frequent Patterns from COVID-19 Streaming Data

Asmaa S. ABDO[1]*, Hatem M. ABDUL-KADER[2], Rashed K. SALEM[2]

[1] Faculty of Computers and Artificial Intelligence, University of Sadat City, Menoufia, 32897, Egypt
asmaa.saad@fcai.usc.edu.eg (**Corresponding author*)

[2] Faculty of Computers and Information, Menoufia University, Menoufia,32511, Egypt
hatem.abdelkader@ci.menofia.edu.eg, rsalem@ci.menofia.edu.eg

Abstract: The Coronavirus disease (COVID-19) pandemic has led to a huge loss of human life. It has also severely affected the economic, social, and health systems around the world. Frequent pattern mining is one of the main research topics in data stream mining. It is significant in many critical applications, especially in the medical field. This paper proposes a Compressed Maximal Frequent Pattern based on a Damped Window model over a data stream (CMFP-DW). Its main contribution is to integrate the concept of correlation with the purpose of finding valuable patterns that are highly correlated. As such, a new type of pattern is defined, namely the correlated compressed maximal frequent pattern. The CMFP-DW approach is employed for mining accurate correlated maximal frequent patterns from streaming data, and it has been validated against a real-world COVID-19 dataset from the healthcare domain. Frequent patterns generated from this dataset are exploited with the purpose of detecting the COVID-19 cases in different countries of the world. This helps decision makers take the appropriate precautions to prevent the further spread of the COVID-19 pandemic across the world. The six experiments carried out show that the proposed approach outperforms two other existing approaches, namely the estDec and the CP-Tree algorithms regarding accuracy in extracting correlated maximal frequent patterns, memory usage, and the required response time.

Keywords: COVID-19, Data stream, Frequent pattern mining, Damped window, Data stream mining, Interestingness measure, Correlation, Bond measure.

1. Introduction

Data stream mining in medical applications such as those related to the COVID-19 pandemic is an interesting topic of research nowadays. COVID-19 was first reported at the end of 2019 in Wuhan (Hebei province, China). It has triggered a serious change in global health systems. Also, it has affected various people causing serious health crises around the world (El-Shafeiy et al., 2020).

Data streams are ordered sequences of items that arrive in time order. In many applications of information systems, the volume of data may be massive and unbounded. Most researchers in recent times focus on mining frequent patterns from data streams. Data mining algorithms over data streams show a trade-off between processing time and accuracy (Borah & Nath, 2017).

Data stream mining is one of the very attractive research areas, which is gaining a lot of importance in various application domains. It links two areas of research, i.e., data mining and data stream (Khine & Win, 2020).

The common algorithms for the data mining process cannot be used directly for data stream mining because data streams have the following characteristics:

- Data stream is a massive and unlimited sequence of data that is continuously generated at a rapid rate;
- Memory utilization for the data stream mining process should be restricted, although new data items are continuously generated in a data stream;
- The newly generated data must be processed in less than a fixed duration to produce an updated data analysis result, to be used immediately on demand (Shin et al., 2014).

Frequent pattern mining is one of the main research topics in the data stream mining due to the higher memory requirements and huge computational costs. It has wide applications in real-world domains such as the medical field, monitoring of patient routines, environmental and weather data, business intelligence management, weblogs, and web page clickstreams (Nasreen et al., 2014; Djenouri et al., 2018).

The current compact representations of Frequent Patterns (FP) are the Closed Frequent Pattern abbreviated as CFP and Maximal Frequent Pattern abbreviated as MFP, which can be utilized in streaming data (Reddy & Govardhn, 2017). The MFP assures a more efficient pattern of compression with additional compact forms in comparison with the CFP representation (Cai et al., 2017). It is a more effective method for enhancing time and space consumption over data streams (Qu et al., 2013; Cai et al., 2020).

https://doi.org/10.24846/v31i1y202210

Different types of window models have been proposed in the literature. Frequent pattern mining from the data stream using window models (Borah & Nath, 2017) are shown in Figure 1. The type of window model is determined by the characteristics of the data streams. In most real-world applications, frequent pattern mining algorithms in data stream prefer to deploy their mining process based on either the damped window model or the sliding window model (Qu et al., 2013; Zhang et al., 2019).

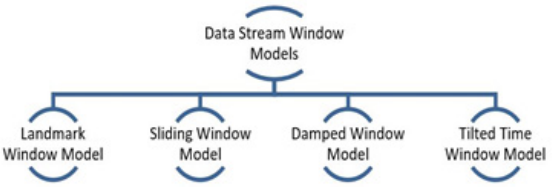

Figure 1. Data stream window models (Borah & Nath, 2017)

This paper approaches the issue of mining correlated maximal frequent patterns. The task is to discover all the correlated maximal frequent patterns from the data stream. This is achieved by using one of the interestingness measures called the bond measure. The bond verifies the importance of the association between items in the same pattern.

First, this paper proposes a Compressed Maximal Frequent Pattern based on Damped Window over a data stream, which is abbreviated as the (CMFP-DW) approach. The proposed CMFP-DW approach uses data stream mining techniques to generate maximal correlated frequent patterns from the data stream. It mainly relies on the extraction of maximal frequent patterns, which speeds up the generation process, especially in the data stream. The resulting maximal frequent patterns generated through the proposed approach are correlated, which aims to enhance their accuracy in comparison with the previous Compressible-prefix tree (CP-tree) algorithm (Shin et al., 2014). The bond measure is one of the interestingness data mining association measures. This measure can be utilized in stream data mining because it investigates the correlation between frequent pattern items. The aim of employing the bond measure is to ensure that the respective items are correlated (Fournier-Viger et al., 2020; Fournier-Viger et al., 2016).

Secondly, the aim of this paper is to enhance memory usage. There is a considerable need to efficiently manage the limited internal memory space in order to store, delete or update unbounded incoming data streams. Experimental results performed on the COVID-19 dataset confirm the efficiency and correctness of the proposed approach against the CP-tree algorithm (Shin et al., 2014). The generated patterns are exploited in many real-world applications, which require reliable, accurate frequent patterns. According to the COVID-19 dataset, the generated frequent patterns are being exploited in order to detect the COVID-19 cases across the world.

The remainder of this paper is structured as follows. Section 2 reviews the related works. Section 3 introduces the proposed CMFP-DW approach for generating correlated maximal frequent patterns. Section 4 discusses the experimental study and results for the selected COVID-19 dataset. Finally, the conclusions and possible future directions are presented in section 5.

2. Related Work

Data stream mining has turned out to be an active area of research interest for database researchers over the last few years. Little research has been done on enhancing semantic meaning and accuracy of generating frequent patterns from the data stream.

The frequent pattern mining methods proposed in traditional data mining do not fit properly in stream data (Zhang et al., 2019). Many traditional data mining methods scan databases multiple times to mine frequent patterns, for example the Apriori algorithm, FP-Growth algorithm, and others (Reddy & Govardhn, 2017). Apriori algorithm scans the transactional database multiple times. FP-Growth algorithm utilizes a compacted FP-tree structure. Both algorithms require a high memory consumption and more than one scan of the transactional database, which is not suitable for data stream mining (Schlegel et al., 2011). In a paper survey (Ramírez-Gallego et al., 2017), the authors summarize the contributions

of pre-processing techniques to data streams. It also refers to both existing algorithms and open challenges related to data streams. Frequent pattern mining approaches and challenges have also been discussed by Borah & Nath (2018). Their paper discusses diverse challenges in streaming data when dealing with dynamic or incremental datasets.

Several interestingness measures have been proposed by the literature on statistics and data mining for evaluating relations between items. That is important in recent research on association or correlation in pattern mining (Wu et al., 2010).

One of the most important research topics of data mining is to obtain dependable results for generating interest patterns from data. Various interestingness measures exist in pattern mining, such as Support, Confidence, All-Confidence, Cosine, lift, F-Measure, bond, etc. (Saraswathi & Nagadeepa, 2018), (Somyanonthanakul & Theeramunkong, 2020). It is very important to select appropriate measures of interestingness according to their role in data mining applications (Sharma et al., 2020), specifically for generating frequent patterns from data streaming. Those generated patterns are used in the various decision-making processes to achieve more reliable results (Afriyie et al., 2020; Kuznetsov & Makhalova, 2018). Bond is one of the most interestingness measures with regard to data mining (Fournier-Viger et al., 2016; Fournier-Viger et al., 2020). It can be utilized in stream data mining because it investigates the correlation between items in the same pattern.

Research in mining patterns from streaming data categorizes them according to three window types: landmark-window-based mining, sliding-window-based mining, and damped-window-based mining (Ghatage, 2015; SiddaReddy et al., 2014).

2.1 Landmark Window Model-based Streaming Approaches

A proposed algorithm called lossy counting is exploited for discovering frequent patterns from streaming data. The algorithm requires a higher amount of time and a large memory space (Manku & Motwani, 2012). The identification of frequent patterns from uncertain data stream mining using the landmark window is discussed by (Leung et al., 2013). The landmark window model may prove to be ineffective, as frequent patterns are highly time sensitive.

2.2 Sliding Window Model-based Streaming Approaches

The proposed Moment algorithm is employed for mining closed frequent patterns, but it works with scanning the dataset multiple times (Chi et al., 2004). In the work of Lee et al. (2014), the authors proposed an algorithm for mining Weighted Maximal Frequent Patterns (WMFPs) over data streams. This algorithm applies weights to the mining process to reflect recent information over the data stream. The major limitation of this algorithm is the amount of time necessary for discovering frequent patterns.

2.3 Damped Window Model-based Streaming Approaches

estDec is the proposed method for data stream mining to adaptively find recent frequent patterns across the online data stream (Chang & Lee, 2003). In the work of Shin et al. (2014), the authors proposed CP-tree for maintaining a compressed prefix tree structure in order to find maximal frequent patterns from streaming data. The major limitations of the CP-tree are still the accuracy of frequent patterns generated and the memory space usage.

2.4 Tilted Time Window Model-based Streaming Approaches

FP-stream is an incremental algorithm for data stream mining. This algorithm maintains the tree data structure in order to represent and discover frequent patterns of the data stream. Several experiments have been performed to prove the effectiveness of this algorithm. The drawback of this algorithm is the high processing time (Borah & Nath, 2017).

Finally, it can be concluded that existing methods do not guarantee that the discovered frequent patterns from streaming data are reliable with regard to the problem of semantic meaning and correlation between items in the dataset. The previous methods employed are not effective in generating frequent patterns from critical data such

as the COVID-19 dataset in the medical healthcare domain. Also, memory and processing time are critical challenges for current data stream mining algorithms. In this paper, the bond measure is a type of interestingness-based data mining measure of association that is utilized for dealing with streaming data, since a frequent pattern having a high bond is considered not just frequent, but also contains items that often co-occur.

3. The Proposed CMFP-DW Approach

The proposed Compressed Maximal Frequent Pattern is based on Damped Window over a data stream, which is abbreviated as the (CMFP-DW) approach. Given a data stream, the proposed approach generates correlated maximal frequent patterns from the data stream. The CMFP-DW approach follows the basic structure of the CP-tree algorithm (Shin et al., 2014). The main steps for generating correlated maximal frequent patterns from the data stream are indicated in the flowchart diagram in Figure 4, as well as in the pseudocode in Figure 5. The proposed approach can be described as follows:

Input: Data stream (Ds) and thresholds, i.e., minimum support threshold (min_supp), minimum significant threshold (min_sign = 0.1 * min_supp), minimum merge threshold (min_merge), merge gap threshold called delta (ð), d (decay rate), and minimum bond threshold (min_bond) are the inputs for the CMFP-DW approach.

Step 1: Given the set of thresholds, the total number of transactions in the current data stream (Ds) is updated by adding new ones.

$$Ds = Ds + 1 \tag{3.1}$$

Step 2: Update the item count of items to the itemsets, and then the tree traversal method by restructuring nodes to lexical order of items. This phase determines the following:

- Updating the decay rate (d) for each item;
- Check items, if min_supp of item < min_sign threshold, then pruning it;
- Determine the splitting or merging method between itemsets based on min_supp, min_merge, and delta (ð) thresholds.

Step 3: The insertion of the set of items for each transaction is performed by updating the item count for each transaction. Then, add them if they are not stored in a tree.

Step 4: The maximal frequent patterns are generated using the compressed prefix tree, which mines the frequent patterns with one dataset scan as shown in Figure 2. Forced pruning is applied periodically to remove infrequent nodes.

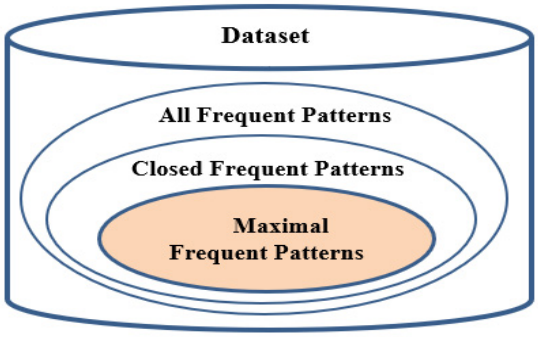

Figure 2. Search space domain (Salem & Abdo, 2016)

Step 5: Generating interestingness-based correlated maximal frequent patterns, by utilizing one of the interestingness data mining measures called bond measure (Fournier-Viger et al., 2016; Fournier-Viger et al., 2020). The bond has a value in the interval [0,1]. A high value means that the itemset is correlated. The goal of using a bond measure is to ensure that the items in an itemset are correlated. The relationship between support and bond measures is illustrated in the diagram in Figure 3.

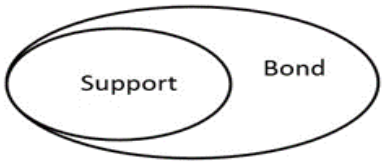

Figure 3. The relationship between support and bond measures

For example, if an itemset from the COVID-19 dataset of the form {Province/State: Hubei, Country/Region: Mainland China, Confirmed: 68135} features a high bond, this usually means its items appear together and are confirmed for 68135 COVID-19 cases. The itemset {Province/State: Hubei, Country/Region: Mainland China, Confirmed: 68135} may appear many times in the dataset, but its items may not be substantially

related, as such, it is advisable to use the bond measure to check this.

$$Bond\ (X) \geq Support\ (X)\ where\ X\ is\ itemset \quad (3.2)$$

The following is the equation of the bond interestingness measure:

$$Bond\,(X) = \frac{Conjunctive\_Support\,(X)}{Disjunctive\_Support\,(X)} \quad (3.3)$$

Conjunctive_Support (X) = how many transactions contain all the items of X together. (3.4)

Disjunctive_Support (X) = how many transactions contain at least one item from X. (3.5)

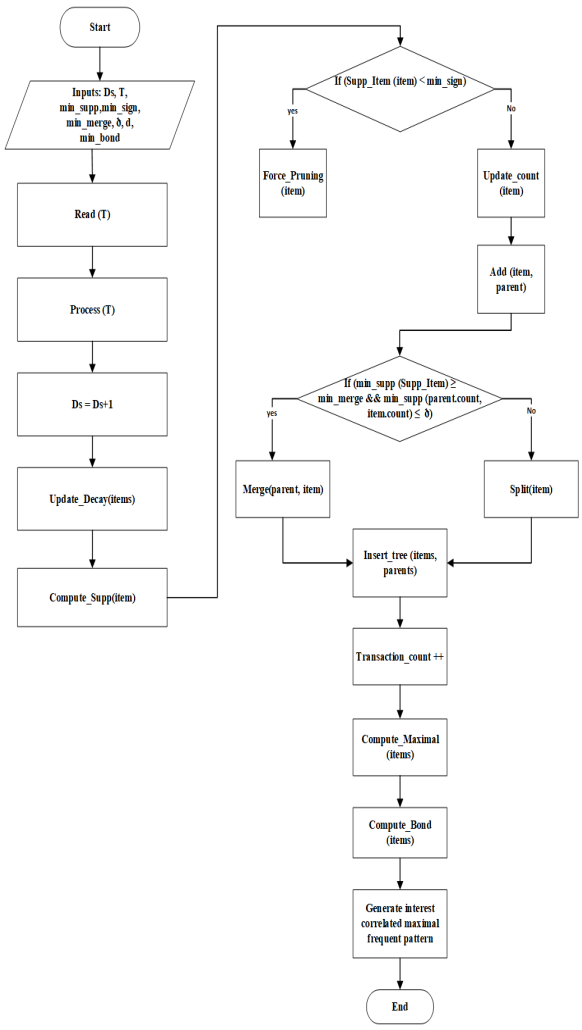

Figure 4. The proposed CMFP-DW flowchart

Pseudo Code of Proposed CMFP-DW

Input: D_S: Data stream
T: Transaction
min_supp: minimum support threshold
min_sign: minimum significant threshold
min_merge: minimum merge threshold
δ: merge gap threshold called delta
d: decay rate
min_bond: minimum bond threshold
Output: Generate interest correlated maximal frequent pattern.

```
1. for each new T in Ds
2.     Process (T) by read it.
3.     Ds=Ds+1;
4.    Update_Decay (items);
5.    Supp_Item ← Compute_Support (items) in T;
6.       If (Supp_Item (item) < min_sign)
7.          Then Force_Pruning (item);
8.          else
9.          Update_count (item);
10.         Add (item, parent);
11.     end if
12.     If (min_supp (Supp_Item) ≥ min_merge
13.                 && min_supp (parent.count, item.count) ≤ δ)
14.       Then merge (parent, item);
15.       else if (min_supp (Supp_Item) ≥ min_merge
16.                         && min_supp (Supp_Item, m) > δ)
17.        Then Split (item);
18.        else;
19.   end if
20.   Insert_tree (items, parents);
21.   Transaction_count ++;
22.   Compute_Maximal (items);
23.   Compute_Bond (items);
24. end for each
```

Figure 5. The proposed CMFP-DW algorithm

4. Experimental Study

An experimental study is presented with the purpose of validating the proposed CMFP-DW approach using a real-world dataset. The performance of the proposed CMFP-DW approach is evaluated for critical applications such as medical applications. The analysed dataset includes a massive amount of daily-level information about the COVID-19 pandemic. This information is related to the number of COVID-19 cases across the globe from January to July 2020. The characteristics of the COVID-19 dataset are shown in Table 1. UTC is the abbreviation for "Coordinated Universal Time".

Table 1. Characteristics of the COVID-19 dataset

ID	Attribute name	Type	Description
1	SNo	Numeric	Serial Number
2	Observation Date	Date/ Time	Observation date as mm/dd/yyyy
3	Province/State	Text	Province or State
4	Country/ Region	Text	Country or Region
5	Last Update	Date/ Time	Last update date/time in UTC
6	Confirmed	Numeric	Cumulative number of confirmed cases
7	Deaths	Numeric	Cumulative number of deaths
8	Recovered	Numeric	Cumulative number of recovered cases

The proposed approach was utilized to generate correlated maximal frequent patterns from this dataset. These patterns become accessible to decision-makers in medical systems and allow them to take accurate decisions.

The following factors are employed in order to evaluate the efficiency and accuracy of the CMFP-DW approach: min_supp, min_sign = (0.1 * min_supp), min_merge, ð, d, and min_bond. For the proposed CMFP-DW approach, d = 1 and min_bond = 0.5 in all experiments. The proposed approach is evaluated by using several measures, including:

- Accuracy in extracting correlated maximal frequent patterns;
- Memory consumption;
- Response time measurement.

4.1 Experimental Setting

Experiments were conducted using a real COVID-19 dataset from (SRK, 2020) namely the Novel Coronavirus 2019 Dataset. This dataset includes around daily information on the number of COVID-19 cases worldwide from January to July 2020 provided by the World Health Organization. The COVID-19 dataset described has a size of 65535 and it includes 8 columns.

The implementation of the proposed approach used Java (JDK1.8) programming. The application was tested on a device of the type Intel(R) Core (TM) i7-8550u CPU @ 1.80 GHz 1.99 GHz. Also, the proposed approach runs in the main memory, it requires 8.00 GB RAM and can run on the Windows 10 operating system. Each experiment that was conducted has been repeated for at least five times, and the average is reported here.

4.2 Experimental Results

This subsection of the experimental study presents and analyzes the results obtained by the proposed approach. Also, the performance and efficiency of the CMFP-DW approach are compared to that of the CP-tree and estDec algorithms (Chang & Lee, 2003; Shin et al., 2014)it is very important to confine the memory usage of a data mining process. This paper proposes a CP-tree (Compressible-prefix tree with regard to mining correlated maximal frequent patterns from the COVID-19 dataset.

Experiment 1

The objective of this experiment is to evaluate the accuracy of generating correlated maximal frequent patterns. It also aims to measure memory space usage. The values of accuracy and memory space usage for frequent patterns generated by the proposed CMFP-DW approach are compared with those obtained for the CP-tree algorithm (Shin et al., 2014) over the COVID-19 dataset. The setting for experiment 1 is indicated in Table 2.

Table 2. The setting for Experiment 1

Setting	Min_supp	Min_sign	Min_merge	Delta
S1	0.01	0.001	0.01	0.01
S2	0.02	0.002	0.02	0.02
S3	0.03	0.003	0.03	0.03
S4	0.04	0.004	0.04	0.02
S5	0.05	0.005	0.05	0.01

Experiment 1-a: Accuracy of generating correlated maximal frequent patterns

This experiment measures the accuracy of generating accurate and dependable maximal correlated frequent patterns. As it is shown in Figure 6, the total number of generated frequent patterns is indicated on the y-axis in case of changing threshold values such as min_supp, min_sign (0.1 * min_supp), min_merge, and delta (ð) on the x-axis. It can be noted that the proposed approach always generates accurate correlated maximal frequent patterns compared to the CP-tree algorithm (Shin et al., 2014).

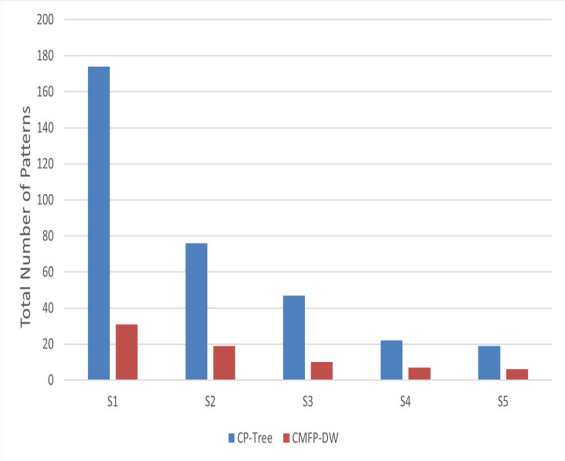

Figure 6. The total number of frequent patterns generated

For example, in Figure 6, at min_supp = 0.01 and min_sign = (0.1*min_supp), the number of patterns generated by the CP-tree algorithm is 174, in comparison with the number of patterns generated by the proposed CMFP-DW approach, that is 31.

Experiment 1-b: Memory space usage for generated patterns

In this experiment, the results from Figure 7 show the memory space usage of patterns generated by the proposed CMFP-DW approach in comparison with the CP-tree algorithm (Shin et al., 2014)it is very important to confine the memory usage of a data mining process. This paper proposes a CP-tree (Compressible-prefix tree. The x-axis includes multi-threshold values such as min_supp, min_sign (0.1*min_supp), min_merge, and delta (ð). The y-axis represents the amount of memory measured in megabytes (MB). This experiment shows that the use of the CMFP-DW approach for generating frequent patterns brings about an improvement in the memory space usage, as it can be seen in Figure 7.

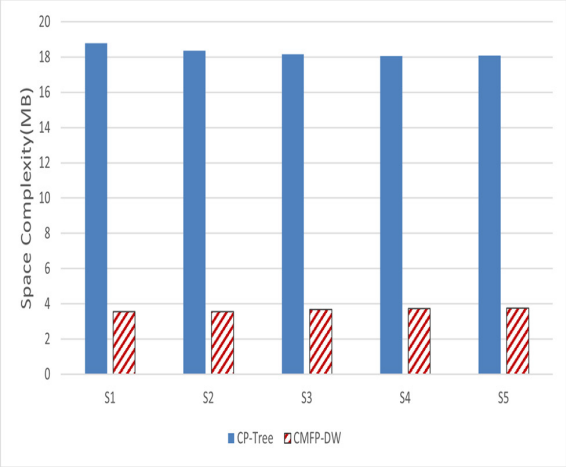

Figure 7. Space complexity measurement with regard to the COVID-19 dataset

Experiment 2: Efficiency in extracting correlated maximal frequent patterns

The efficiency of the proposed CMFP-DW approach in extracting correlated maximal frequent patterns is analyzed in comparison with that of the CP-tree algorithm (Shin et al., 2014). The results for the selected dataset are shown in Figure 8. In case of changing the value of the min_supp threshold, Figure 8 shows the efficiency of the proposed CMFP-DW approach in reducing the number of correlated maximal frequent patterns generated in comparison with the CP-tree algorithm (Shin et al., 2014). As it is shown in Figure 8, different values are set for the min_supp threshold and the values of other parameters are set as follows: min_sign = (0.1 * min_supp), min_merge = 0.05, delta (ð) = 0.01, and bond = 0.5. It can be noted that when increasing the min_supp threshold, the number of correlated maximal frequent patterns is decreasing.

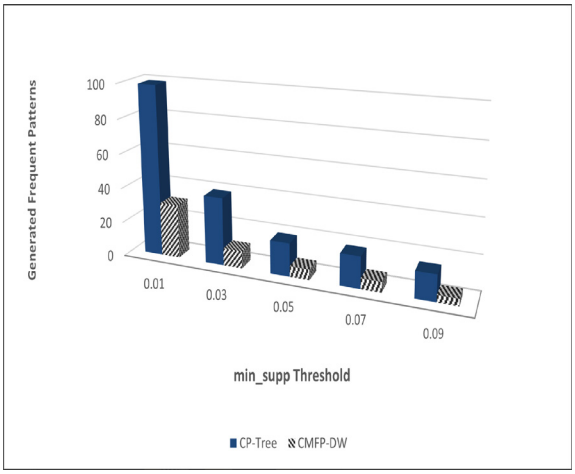

Figure 8. Correlated maximal frequent patterns

Experiment 3: Number of nodes generated for different values of the ð threshold

This experiment determines the total number of generated nodes by changing the values of the delta (ð) threshold, as it is illustrated in Figure 9.

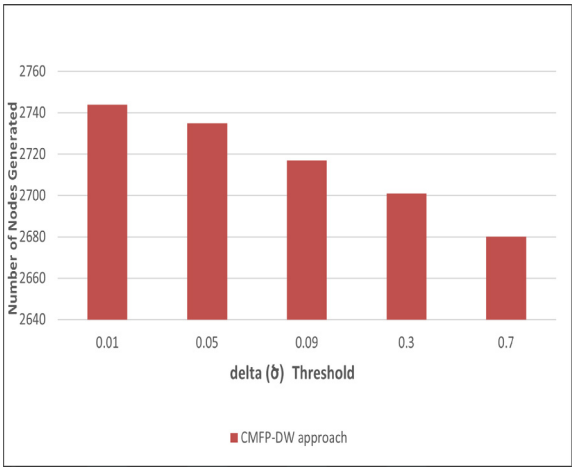

Figure 9. The number of nodes generated with different values of the delta (ð) threshold

The delta (ð) ∈ (0,1) and the values of other parameters are set as follows: min_supp = 0.03, min_sign = (0.1*min_supp), min_merge = 0.03, and bond = 0.5. Based on the results obtained, it can be concluded that by increasing the value of delta (ð), more nodes can be merged. It is recommended to the value of the delta (ð) threshold in order to enhance the accuracy of generated correlated frequent patterns. As it is shown, when delta (ð) = 0.01, the total number of nodes amounts to 2744 nodes. By increasing the value of the min_merge threshold, more nodes can be merged. For example, if delta (ð) = 0.7, the total number of nodes amounts to 2680.

Experiment 4: Memory space usage for different values of the min_merge threshold

The experiment aims to measure memory usage for the proposed CMFP-DW approach if the min_merge threshold is changed, as it is shown in Figure 10. The values of other parameters are set as follows: the value of delta (ð) and min_supp is fixed for 0.01, min_sign = (0.1 * min_supp), bond = 0.5, and min_merge threshold ∈ (0,1). As it is shown in Figure 10, the min_merge takes different values. As more nodes are merged, memory space usage is also reduced. When the min_merge value is higher, fewer nodes are merged, and the amount of memory usage increases.

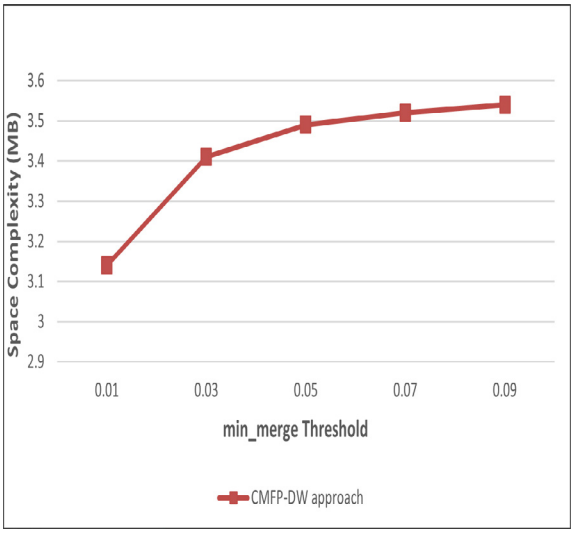

Figure 10. Measuring memory usage for different values of the min_merge threshold

Experiment 5: Measuring response time for different values of the min_merge threshold

This experiment determines the required processing time in the case of different min_merge values, as it is shown in Figure 11. The response time decreases as the values of the min_merge threshold increase. The proposed approach performs better because the increase of the value of the min_merge threshold reduces the merging between nodes. The min_merge ∈ (0,1) and the values of other parameters are set as follows: min_supp = 0.01, min_sign = (0.1*min_supp), delta (ð) = 0.01 and bond = 0.5.

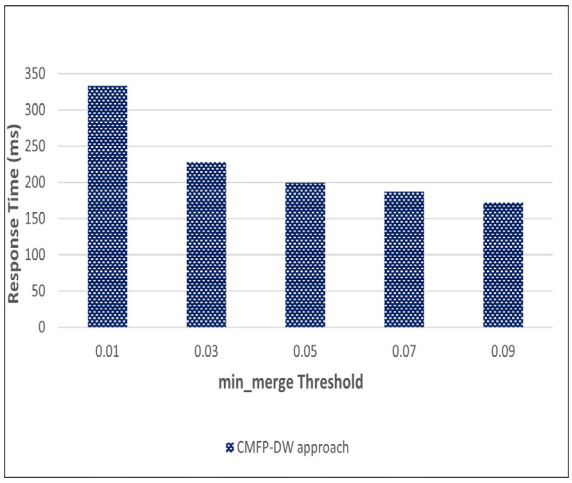

Figure 11. Measuring response time for different values of min_merge threshold

Experiment 6: Comparison between estDec, CP-Tree, and CMFP-DW

This experiment aims to compare the proposed CMFP-DW and two other existing approaches, namely estDec (Chang & Lee, 2003), and CP-Tree (Shin et al., 2014). The three algorithms use damped window model streaming. As it is shown in Figure 12, the proposed approach outperforms the other two algorithms with regard to saving the memory space.

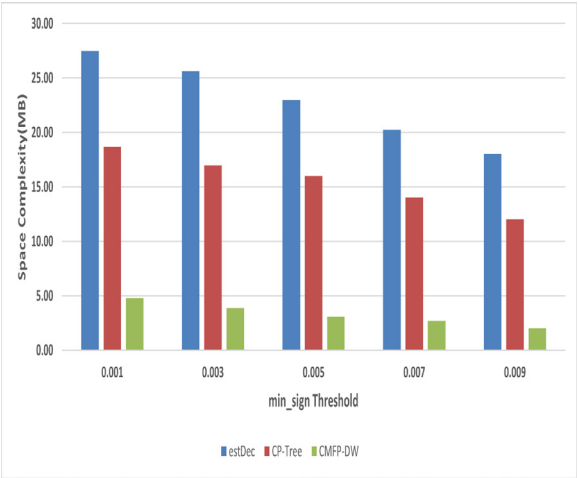

Figure 12. Memory usage for estDec, CP-Tree and CMFP-DW

Discussion of results

Experiments with the proposed approach show that it always produces a smaller number of frequent patterns, but these patterns are more accurate. The generated frequent patterns are correlated with maximal frequent patterns. Experiments also show that the CMFP-DW approach outperforms the other two algorithms with regard to the space complexity of the frequent patterns that are generated. This reduction leads to a better performance with respect to memory usage and response time. In all experiments, the transactions related to the COVID-19 dataset were inserted one by one in the sequence of data in order to simulate the online data stream environment.

5. Conclusion

Due to time and memory constraints, some data preprocessing techniques are required to access each data element at most once. This paper presents a compressed maximal frequent pattern based on a damped window model over a data stream (CMFP-DW). The CMFP-DW approach generates correlated maximal frequent patterns using one of the measures of interest, called a bond. The CMFP-DW approach is validated and evaluated on a real-world COVID-19 medical dataset. The experimental results validate the efficiency of the CMFP-DW approach against the CP-tree algorithm. The CMFP-DW approach obtains good results with regard to various aspects such as accuracy in extracting correlated maximal frequent patterns, memory usage, and response time measurement. Finally, the aim is to propose an approach for mining data streams by using concept drift.

Acknowledgments

The authors would like to thank Dr. Engy El-Shafeiy (University of Sadat City) for his support and motivation during the research that has been carried out.

REFERENCES

Afriyie, M. K., Nofong, V. M., Wondoh, J. & Abdel-Fatao, H. (2020). Mining Non-redundant Periodic Frequent Patterns. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12033 LNAI*. Springer.

Borah, A. & Nath, B. (2017). Mining patterns from data streams: An overview. In *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017* (pp. 371-376).

Borah, A. & Nath, B. (2018). FP-Tree and Its Variants: Towards Solving the Pattern Mining Challenges. In *Proceedings of First International Conference on Smart System, Innovations and Computing* (pp. 535-543), Springer, Singapore.

Cai, S., Li, L., Li, S., Sun, R. & Yuan, G. (2020). An efficient approach for outlier detection from uncertain data streams based on maximal frequent patterns, *Expert Systems with Applications*, *160*, 113646.

Cai, S., Sun, R., Cheng, C. & Wu, G. (2017). Exception detection of data stream based on improved maximal frequent itemsets mining, *Communications in Computer and Information Science*, *704* (pp. 112-125).

Chang, J. H. & Lee, W. S. (2003). Finding recent frequent itemsets adaptively over online data streams. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 487-492).

Chi, Y., Wang, H., Yu, P. S. & Muntz, R. R. (2004). Moment: Maintaining closed frequent itemsets over a stream sliding window. In *Proceedings of the Fourth IEEE International Conference on Data Mining, ICDM 2004* (pp. 59–66).

Djenouri, Y., Belhadi, A. & Fournier-Viger, P. (2018). Extracting useful knowledge from event logs: A frequent itemset mining approach, *Knowledge-Based Systems*, *139*, 132-148.

El-Shafeiy, E., Hassanien, A. E., Sallam, K. M. & Abohany, A. A. (2020). Approach for training quantum neural network to predict severity of COVID-19 in patients, *Computers, Materials and Continua*, *66*(2), 1745-1755.

Fournier-Viger, P., Lin, J. C. W., Dinh, T. & Le, H. B. (2016). Mining correlated high-utility itemsets using the bond measure. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 53-65) Springer, Cham.

Fournier-Viger, P., Zhang, Y., Lin, J. C. W., Dinh, D. T. & Le, H. B. (2020). Mining correlated high-utility itemsets using various measures, *Logic Journal of the IGPL*, *28*(1), 19-32.

Ghatage, R. A. (2015). Frequent Pattern Mining Over Data Stream Using Compact Sliding Window Tree & Sliding Window Model, *International Research Journal of Engineering and Technology*, *2*(4), 217-223.

Khine, P. T. T. & Win, H. P. P. (2020). Ensemble Framework for Big Data Stream Mining. In *2020 IEEE Conference on Computer Applications, ICCA 2020* (pp. 1-5).

Kuznetsov, S. O. & Makhalova, T. (2018). On interestingness measures of formal concepts, *Information Sciences*, *442-443*, 202-219.

Lee, G., Yun, U. & Ryu, K. H. (2014). Sliding window based weighted maximal frequent pattern mining over data streams, *Expert Systems with Applications*, *41*(2), 694-708.

Leung, C. K. S., Cuzzocrea, A. & Jiang, F. (2013). Discovering frequent patterns from uncertain data streams with time-fading and landmark models. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems VIII: Vol. 7790 LNCS* (pp. 174-196).

Manku, G. S. & Motwani, R. (2012). Approximate frequency counts over data streams. In *Proceedings of the VLDB Endowment*, *5*(12), 1699.

Nasreen, S., Azam, M. A., Shehzad, K., Naeem, U. & Ghazanfar, M. A. (2014). Frequent pattern mining algorithms for finding associated frequent patterns for data streams: A survey, *Procedia Computer Science*, *37*, 109-116.

Qu, Z. G., Niu, X. X., Deng, J., McArdle, C. & Wang, X. J. (2013). Frequent itemset mining over stream data: Overview. In *IET International Conference on Information and Communications Technologies (IETICT 2013)*, (pp. 35-40).

Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M. & Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions, *Neurocomputing*, *239*, 39-57.

Reddy, V. S., Narendra, M. & Helini, K. (2014). Knowledge Discovery from Static Datasets to Evolving Data Streams and Challenges, *International Journal of Computer Applications*, *87*(15), 22-25.

Reddy, V. S., Rao, T. V. & Govardhn, A. (2017). CASW: Context Aware Sliding Window for Frequent Itemset Mining over Data Streams, *International Journal of Computational Intelligence Research*, *13*(2), 183-196.

Salem, R. & Abdo, A. (2016). Fixing rules for data cleaning based on conditional functional dependency, *Future Computing and Informatics Journal*, *1*(1-2), 10-26.

Saraswathi, P. & Nagadeepa, N. (2018). Extracting the significant rules using interestingness measures in mining techniques, *International Journal of Pure and Applied Mathematics*, *118*(5), 493-497.

Schlegel, B., Gemulla, R. & Lehner, W. (2011). Memory-Efficient Frequent-Itemset Mining. In *Proceedings of the 14th International Conference on Extending Database Technology* (pp. 461-472).

Sharma, R., Kaushik, M., Peious, S. A., Yahia, S. Ben & Draheim, D. (2020). Expected vs. Unexpected: Selecting Right Measures of Interestingness. In *International Conference on Big Data Analytics and Knowledge Discovery* (pp. 38-47). Springer.

Shin, S. J., Lee, D. S. & Lee, W. S. (2014). CP-tree: An adaptive synopsis structure for compressing frequent itemsets over online data streams, *Information Sciences*, *278*, 559-576.

SRK (2020). *Novel Corona Virus 2019 Dataset*, SRK. Available at: <https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset>, last accessed: 21 February 2022.

Somyanonthanakul, R. & Theeramunkong, T. (2020). Characterization of interestingness measures using correlation analysis and association rule mining, *IEICE Transactions on Information and Systems*, *103*(4), 779-788.

Wu, T., Chen, Y. & Han, J. (2010). Re-examination of interestingness measures in pattern mining: A unified framework, *Data Mining and Knowledge Discovery*, *21*(3), 371-397.

Zhang, Z., Chen, J., Chen, L., Liu, Q., Yang, L., Wang, P. & Zheng, Y. (2019). A scalable method of maintaining order statistics for big data stream, *Computers, Materials and Continua*, *60*(1), 117-132.