CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

Frequent pattern extraction from the trajectory data suffered by the various problems (complexity, memory consumption, etc..) due to the technical advancements and real-time applications. The continuous processing of data and the too massive data stream are the major causes for those problems. In this thesis, the challenges in frequent pattern mining from the trajectory data. Dynamic and the continuous evolving nature of Transaction Database (TD) required the up-to-date analysis that leads to the development of new work to overcome the complexities.

The presence of contender entities in the TD reduced the mining performance with more time and memory consumption. These problems are addressed in this thesis with the methods available traditionally and proposed the tree-based space partition approach to overcome the limitations. UP-Growth+ algorithm and BM Search for mining the service entities in the trajectory database. The performances of the UP-Growth+ algorithm and BM search algorithm are compared with the metrics such as time, memory, accuracy and data reduction. The comparison results showed that the time consumption, memory consumption, and the percentage of CPU usage, accuracy, and data reduction rate of the suggested BM search algorithm and UP-Growth algorithm are 1300 ms, 38 MB, 14%, 83%, and 84% respectively. Whereas, they 3100 ms, 12MB, 16%, 71%, and 70% respectively. UP-Growth algorithm and BM Search algorithm are implemented for many of the service entities in the trajectory databases. Thus, from the comparative analysis between the UP-Growth+ and BM
search the research work proved that BM search algorithm returned the optimal results for all the metrics.

The proposed TSPTPM is developed to find the trajectory frequent mining. The TSPTPM algorithm is used for performing the frequent pattern mining in real-time diverse datasets such as chess, mushroom, connect and accident. The implementation of TSPTPM included two major modules such as heap tree formation and vague space partition. Initially, the transactions from the database are clustered together with ID. Then, the cluster is passed to the heap tree formation module to construct the tree. The transaction with maximum ID is considered as the root node and the child nodes are placed right and left to the root node based on the count value of transactions (even and odd). Then, vague space partition is applied to the heap tree to extract the frequent patterns. The items demanded for the cluster formation from the selected datasets are 500, 1000, 1500, 2000, 2500 and 3000. The analysis of mining time with respect to the chess dataset shows that the proposed TSPTPM algorithm minimizes the mining time by 34% for 500 cluster items, 52% for 1000 cluster items, 43% for 1500 cluster items, 36% for 2000 cluster items, 19% for 2500 cluster items and 16% for 3000 cluster items. Similarly, the analysis of mining time for the mushroom dataset shows that the proposed TSPTPM algorithm minimizes the mining time by 64% for 500 cluster items, 61.5% for 1000 cluster items, 55.8% for 1500 cluster items, 49% for 2000 cluster items, 5.3% for 2500 cluster items and 8% for 3000 cluster items. When compared to the existing Apriori algorithm, the proposed TSPTPM algorithm decreases the mining time for the connect dataset by 64% for 500 cluster items, 61.5% for 1000 cluster items, 55.8% for 1500 cluster items, 49% for 2000 cluster items, 5.3% for 2500 cluster items and 8% for 3000 cluster items. Similarly, for the accident dataset, the mining time of the TSPTPM algorithm
is reduced by 43.8% for 500 cluster item, 25.6% for 1000 cluster items, 26% for 1500 cluster items, 15.9% for 2000 cluster items, 16.8% for 2500 cluster items and 31.7% for 3000 cluster items. Thus, when compared to the existing Apriori algorithm, the proposed TSPTPM algorithm provides minimal mining time for all the datasets.

The analysis of mining time with respect to the support value shows that the proposed TSPTPM algorithm consumes minimal mining time for all the support values. The analysis of memory consumption for the existing IBT, DBV, CHARM and BSM algorithms and the proposed TSPTPM algorithm shows that the proposed algorithm consumes minimal memory irrespective of the support value. Further, the number of rules generated for the existing IBT, DBV, CHARM and BSM algorithms and the proposed TSPTPM algorithm shows that the proposed TSPTPM algorithm minimized the number of rules generated.

6.2 FUTURE ENHANCEMENT

As the location prediction demands more number of pattern retrieval, the time consumed for the retrieval process increases. Thus, to enhance the performance of the TSPTPM, the pattern can be updated in the cloud environment. The proposed TSPTPM provides to enhance the trajectory frequent mining for data obtain from Global Position System(GPS) in future. The TSPTPM algorithm may have used to identify frequent trajectory according to the movement object in specific area. The computational time of the TSPTPM can be minimized by implementing the concept of parallel computation in order to return the better results.