sup(de) = 1. Secondly, MDIAR-SW generates one candidate 3-itemset (bde) and uses bitwise AND operation to count the sup(bde) = 1, i.e., Bit(bd) AND Bit(be) = 0010. The 3-itemset (bde) is infrequent. Because no new candidates are generated, the generation of frequent and infrequent itemset process is stopped. Hence, there are five frequent itemsets, (b), (d), (e), (bd), (be), infrequent itemsets, (c), (bde), generated by MDIAR-SW algorithm in TransSW2.

V . Experiment

In this section, we evaluate the performance of our proposed algorithm for mining indirect temporal sequential patterns. The computation environments are i5-3470, 4G RAM, Windows 7 operating system. The algorithm is implemented with C++. The synthetic experiment data set is generated by Assocgen[4].

The synthetic data stream, denoted as T5I4D1000K, of size 1 million transactions (D1000K) has an average transaction size of 5 items (T5) with average maximal frequent itemset size of 4 items (I4). In the experiments, the transactions of T5I4D1000K are looked up in sequence to simulate the environment of an online data stream.

The size of a sliding window w, the minimum support threshold s itempair support threshold ts , mediator support threshold tf, dependence threshold and td are set to 20,000, 0.1% ,0.1%, 0.2% and 50%, respectively. As shown in these experiments, the processing times of MDIAR-SW algorithm are shown in Figures 1 and 2.

Fig.1 shows the processing time of window initialization phase under different window sizes from 20,000 (200K) transactions to 100,000 (1,000K) transactions. Fig.2 shows the total time of window sliding time and pattern mining time at each 100K transactions using various window sizes from 200K transactions to 1000K transactions. As shown in Figures 1 and 2, MDIAR-SW algorithm.

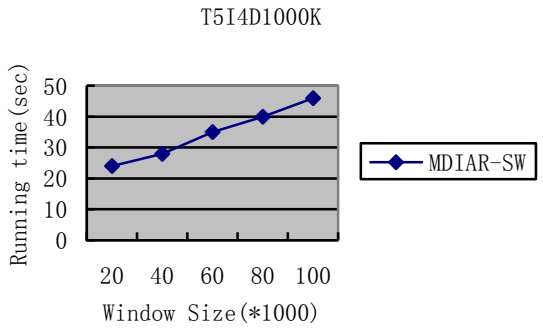

Fig.1 Running time in window initialization phases of algorithm MDIAR-SW under different window size.

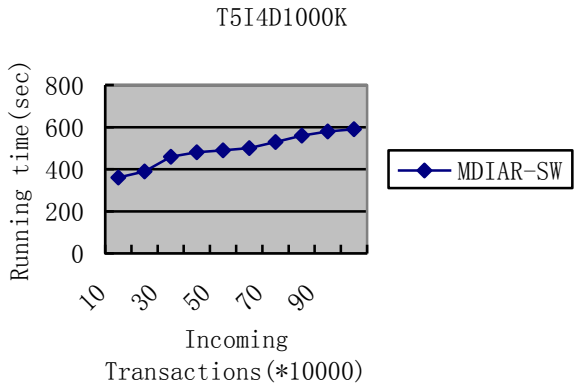

Fig.2 Running time including window sliding time and rule generation time of algorithm MDIAR-SW under different window size 200K transactions.

VI . Conclusions and Future Works

In this paper, we proposed an efficient one-pass algorithm, called MDIAR-SW, for mining direct and indirect association rules over online data streams with a sliding window. Experiments show that the proposed algorithm is efficient and scalable.

References

[1] N. Jiang, and L. Gruenwald. Research Issues in Data Stream Association Rule Mining. In SIGMOD Record, Vol. 35, No. 1, Mar. 2006.

[2] Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems. In Proc. of the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.Wisconsin: Madison, 2002: 1-16.

[3] Agrawal R,Srikant R. Fast Algorithms for Mining Association Rules In the Proc. of the 20th International Conference on VLDB. Santiago, 1994. pp.487~499.

[4] Agrawal R,Srikant R. Mining sequential patterns. In the Proc.1995 Int Conf. on Data Engineering, Taibei,Taiwan,March 1995,pp3-14.

[5] P.N.Tan and V.Kumar. Indirect Association: Mining Higher Order Dependences in Data. Proc. Of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases,pp632-737,Lyon,France(2000).

[6] Y. Chi, H. Wang, P. Yu, and R. Muntz. MOMENT: Maintaining Closed Frequent Itemsets over a Stream Sliding Window. In Proceedings of the 4th IEEE International Conference on Data Mining, pp. 59-66, 2004.

[7] C.H. Lin, D.Y. Chiu, Y.H. Wu and A.L.P. Chen. Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window. In Proceedings of 2005 SIAM International Conference on Data Mining, 2005.