International Journal of Recent Trends in Electrical & Electronics Engg., Feb. 2014.
©IJRTE ISSN: 2231-6612

# An Effective Approach for Detecting Abnormal Activity in Temporal Activity Models

G. Karthika[1], V. Bhuvaneshwari[2]
[1]PG Student, [2]Assistant Professor
Department of Computer Science,
Sri Vidya College of Engineering and Technology, Tamil Nadu, India,

***ABSTRACT***

*Nowadays, many process has to be done in the internet such as online shopping, online ticket booking etc. During this process, timer has to be maintained to find the hacking process. This paper address the identification problem that does not detecting the variation of time values of real time data and non-real time data in a fast manner. To find the time variation of two activities, propose an active comparison algorithm to perform comparisons of real time data and non-real time data fastly. Based on result produced from the active comparison algorithm, and then find the normal and abnormal activity (anomalous activity).*

***INDEX TERMS-*** *Abnormal Activity detection, time stamped data, real time data, on-real time data, and temporal activity.*

## I. INTRODUCTION

There are number of applications we need to monitor whether certain (normal or abnormal) activities are occurring within a flow of transaction data. For example, an online shop might want to monitor the activities occurring during a remote login session on its Web site in order to either better help the user or to identify users engaged in suspicious activities. In existing work, t MAGIC-id algorithm has to be used for identification problem that means matching of real and non-real time data. But it does not perform comparisons of real data and experimental data fastly. Active Comparison algorithm has to be performed comparisons fastly. Temporal activity means that activity related with time. We have to mainly providing security for some applications such as bank and ATM systems. We have to provide security by detecting the abnormal activity by using the time variations. If abnormal activity has to be detected automatically system has to be sending an alert message to the admin.

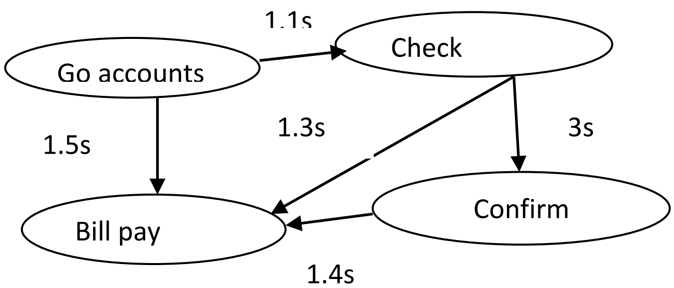


**Fig 1.**Example for Temporal Activity

In above fig, for each activity has different time values. So time calculation has to be calculated separately for each activity. In this paper, online ticket booking process has to be done. Time calculation has to be calculated only for particular webpage. In that webpage, only process for entering banking details such as account number and pin number. Timer has to be maintained only for that particular webpage. By using the time values of the particular webpage, we have to detect the abnormal activity. Abnormal activity means that time value is beyond the limit. Limitations of time values have to be collected from the experimental data (non-real time data).

Experimental data or non-real time data has to be collected by previously performing experiments. That dataset has to be collected from bank sector. Bank sector has to provide the time limitations for entering the bank details. Bank sector and online ticket booking seller has dealing in background. Both have to be deal with the time limitations by experimenting some previous online ticket booking process.

Normally, traffic has to be occurred in network due to large number of persons working at the same time. At that time some delay will be occurred. So hacker has to be easily taking the user's account number and pin number. After all the webpage details have to be filled by the user, they have to submit the online ticket booking process, but still some delay has to be occurred. To avoid this delay, they planned, some time variation is high (beyond the specified time limitations) in between process (entering bank detail process), and automatically some alert message has to be send to the admin.

That alert message contains the time value is beyond the limit, so stop the whole process and check if any hacking process is going on. Immediately admin has to stop the ticket booking process temporarily. After verifying the reason for delay, then start the ticket booking process.

# II. RELATED WORK

Limitations of traditional database management systems in supporting streaming applications and event processing have prompted extensive research in Data Stream Management Systems (DSMSs). Early yet comprehensive survey of relevant issues in data stream management was presented in [6]. Amongst the several systems resulting from research efforts in this direction, of particular relevance is Telegraph CQ [4], a streaming query processor that filters, categorizes, and aggregates flow records according to one or more CQL [2] continuous queries, generating periodic reports.

A significant portion of research in this area has been devoted to optimization of continuous queries [7]. Other works target the recognition of events based on streams of possibly uncertain data [8].Although the system we propose in this paper operates on streams of observation data, the scope of our work is drastically different from the scope of DSMSs.

The aim of past work on indexing of activities was merely to retrieve previously recognized activities, not to recognize new ones. Such work includes that of Ben-Arie et al. [2] who use multidimensional index structures to store body pose vectors in video frames. In short, past work does not address the issue of indexing observations to find activity instances—more importantly, these indexing approaches do not account for uncertainty in what defines an activity which is key to any HMM or stochastic automaton based definition of activities. For instance, Duong et al. [5] introduce the Switching Hidden Semi-Markov Model, a two-layered extension of the Hidden Semi-Markov Model (HSMM).A survey of temporal concepts and data models used in unsupervised pattern mining from symbolic temporal data is presented in [10]. A probabilistic extension of Petri nets for activity detection is proposed in [1]. In conclusion, our work differs from previous efforts by providing a algorithm to perform identification fastly.

## 2.1 Active Comparison Algorithm

An **active comparison** algorithm is a type of comparison algorithm that compares the elements fastly with operations such as "less than or greater than".

It compares all data types such as integers, Booleans, floating-point numbers, characters and alphanumeric strings.

```
Function element Compare (e1, e2)
Count ← 0
N ← 0
For Count<=N
If e1<=e2 Then
Return e1
Else
If e1>=e2 Then
Return e2
End if
```

N=N+1
Return N
End for

**Fig 2.**Active Comparison algorithm

In this above fig, it shows that element Compare is the function to perform comparison operations. It compare two elements such e1 and e2.e1 is the real time data and e2 is the non-real time data. Then 'Count' and 'N' are initialized to zero. Count value is constant because the process never stopped whereas online ticket booking process has to be done continuously is the variable which indicates that how much process has to be done. Then compare Count and N value but Count Value is always less than N value. So process inside the for loop has to be done continuously. Process such as comparing of two elements e1 and e2 with two operations "less than or greater than". If e1 (real time data) is less than e2 (non-real time data) then return the real time data because specified value of non-real time data has some limited time value. So beyond the time value means it has to be considered as that value related to the activity is abnormal activity.
If e1 (real time data) is greater than e2 (non-real time data) then return the non-real time data. So it is below or equal to the time value means it has to be considered as that value related to the activity is normal activity. After that above two processes has to be done, it has to increment the 'N' value and then return the N value. Running time has to be improved by using Timeframe Pruning strategy. This strategy has to be explained in result analysis with graph.

## III. PROBLEM DESCRIPTION

This paper does not solve identification problem that means matching of real data and experimental data (non-real time data) fastly. Goal is to find the abnormal activity.Both non-real and real data does not evaluate running time for the identification problems. Timeframe Pruning strategy has to improve the running time.

## IV. SYSTEM PROCESS

Initially, input data has to be given to the system, then loading the non-real time dataset in the system. Next compare the real time and non-real time data. Non-real Dataset related to the experimental time values for entering the bank details such as account number and pin number. Dataset consists of 20 activities time values such as bkt1.Another dataset related with bus details such as Bs1.Bs1 dataset consists of 30 bus types and its details.

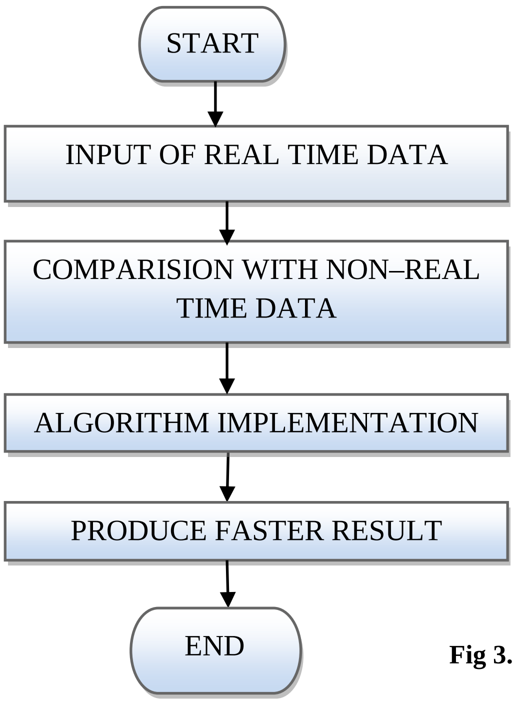


**Fig 3.** Overall System Process

In above fig, during comparison process, Active Comparison algorithm has to be implemented. It produce a faster result by using Timeframe pruning strategy.

## V. Result Analysis

This section describes experiments on both non-real and real data to evaluate running time for the identification problems described in the paper. Adding some strategy for evaluate running time such as TF(Timeframe Pruning). It represents specified period of time in which something occurs or is planned to take place and it removes unwanted time delay to perform the action.

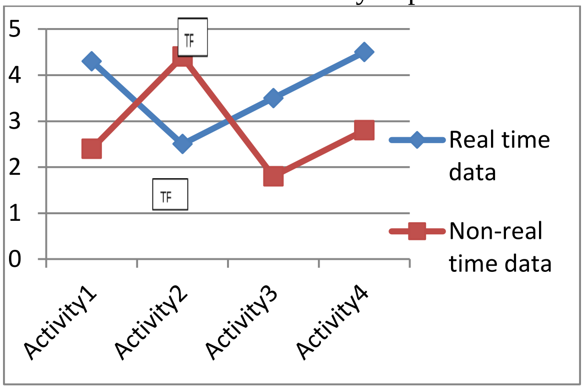


**Fig 4.**Identification process when varying time variations of real and non-real time data

| Activity id | Activity name | Time sec (non Real time data) | Time sec (real-time data) | Identification |
|---|---|---|---|---|
| 1 | Activity1 | 3s | 3.3s | Abnormal activity |
| 2 | Activity2 | 4s | 4s | Normal Activity |
| 3 | Activity3 | 5s | 5.1s | Abnormal Activity |
| 4 | Activity4 | 6s | 6s | Normal Activity |

It is clear that the application of TF improves running time. We then used a third party real-world data set to validate the results obtained on non-real data.

**Fig 5.** Identification process example

We perform experiments used a third party proprietary real-world travel data set including events such as online ticket booking system

## VI. Conclusion

In this paper, we solve the identification problem that does not detecting the variation of time values of real time data and non-real time data in a fast manner**.** Active Comparison algorithm has to be used to solve identification problem fastly. Proposed algorithm mainly focuses on improve the speed of query execution and perform comparison of time values fastly. Based on result produced from the Active comparison algorithm, we have to find the normal and abnormal activity. It gives a security for many applications, mainly in online shopping.

## References

[1] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V.S.Subrahmanian, P. Turaga, and O. Udrea, “A Constrained Probabilistic Petri Net Framework for Human Activity Detection in Video,” IEEE Trans. Multimedia, vol. 10, no. 8, pp. 1429-1443,Dec. 2008.

[2] A. Arasu, S. Babu, and J. Widom, “The CQL Continuous Query Language: Semantic Foundations and Query Execution,” Int’lJ. Very Large Data Bases, vol. 15, pp. 121-142, June 2006.

[3] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, “Human Activity Recognition Using Multidimensional Indexing,” IEEE Trans.Pattern Analysis Machine Intelligence, vol. 24, no. 8, pp. 1091-1104,Aug. 2002.

[4] S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M.Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F.Reiss, and M.A. Shah, “TelegraphCQ: Continuous Dataflow Processing for an Uncertain World,” Proc. Conf. Innovative Data Systems Research (CIDR ’03), 2003.

[5] T.V. Duong, H.H. Bui, D.Q. Phung, and S. Venkatesh, “Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model,” Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR ’05), 2005.

[6] L. Golab and M.T. O ¨ zsu, “Issues in Data Stream Management,”ACM SIGMOD Record, vol. 32, pp. 5-14, June 2003.
[7] F. Reiss, K. Stockinger, K. Wu, A. Shoshani, and J.M. Hellerstein,“Enabling Real-Time Querying of Live and Historical Stream Data,” Proc. 19th Int’l Conf. Scientific and Statistical Database Management (SSDBM ’07), 2007.
[8]S. Wasserkrug, A. Gal, O. Etzion, and Y. Turchin, “Complex Event Processing over Uncertain Data,” Proc. Second Int’l Conf. Distributed Event-Based Systems (DEBS ’08), pp. 253-264, 2008.
[9]B. Kerkez, “Learning Plan Libraries for Case-Based Plan Recognition,”Proc. Midwest Artificial Intelligence and Cognitive Science Conf.
(MAICS ’02), Apr. 2002.
[10]F. Mo¨ rchen, “Unsupervised Pattern Mining from Symbolic Temporal Data,” SIGKDD Explorations Newsletter, vol. 9, no. 1, pp. 41-55, June 2007.