ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Statistical Learning for Anomaly Detection in Cloud Server Systems: A Multi-Order Markov Chain Framework

Pradeep Vajja
Department of Computer Science & Engineering, Sarada Institute of Science, Technology and Management, Srikakulam, Andhra Pradesh 532404, India.

Mr.Simma Seshagiri
Department of Computer Science & Engineering, Sarada Institute of Science, Technology and Management, Srikakulam, Andhra Pradesh 532404, India.

ABSTRACT
As a major strategy to ensure the safety of IT infrastructure, anomaly detection plays a more important role in the cloud computing platform which hosts the entire applications and data. On top of the classic Markov chain model, we proposed in this paper a feasible multi-order Markov chain-based framework for anomaly detection. In this approach, both the high-order Markov chain and multivariate time series are adopted to compose a scheme described in algorithms along with the training procedure in the form of statistical learning framework. To curb time and space complexity, the algorithms are designed and implemented with a non-zero value table and logarithm values in initial and transition matrices. For validation, the series of system calls and the corresponding return values are extracted from classic Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation data set to form a two-dimensional test input set. The testing results show that the multi-order approach can produce more effective indicators: in addition to the absolute values given by an individual single-order model, the changes in ranking positions of outputs from different-order ones also correlate closely with abnormal behaviors.

Key words: ***IT infrastructure, anomaly detection, the high-order Markov chain, multivariate time series, Defense Advanced Research Projects Agency (DARPA), multi-order approach***

INTRODUCTION
Cloud computing [1] comes with indispensable dependency on networked computer systems. Unfortunately, while everyone knows there is no guarantee of its well-being, we tend to simply ignore this painful idea. An increasing number of academical and industrial users are starting to rely solely on cloud computing servers that host entire applications and storage. Thus, taking care of both business and personal data, servers [3] expose critical safety and availability issues. Their invulnerability is of major importance to both individuals and society. However, during catastrophic disasters such as intrusion, crash or breakdown, the anomalies must be first discovered before any actual remedy could come to its aid. Being recessive at the early stage, such problems would not exhibit distinct traits and often lead to delayed responses and irrecoverable results. These approaches are usually categorized into three groups, i.e. statistical approaches, machine learning approaches [5], and data mining approaches: In statistical approaches, anomaly or intrusion detection systems usually watch behaviors of observed objects to comprise statistical distributions as a set of trained profiles during the training phase. These systems then apply the set of trained profiles by comparing them against a new set of profiles of observed objects during the detection phase. Machine learning-based approaches tend to reduce the supervision costs during the training phase of statistical approaches by enabling machine learning-based systems to learn and improve their performance on their own. Neural

Cite this article as: Pradeep Vajja & Mr.Simma Seshagiri, "Statistical Learning for Anomaly Detection in Cloud Server Systems: A Multi-Order Markov Chain Framework", International Journal of Research in Advanced Computer Science Engineering, Volume 5 Issue 4, 2019, Page 1-7.

ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

networks [7] and Hidden Markov Model have been proved to be a useful technique. Neural networks and Hidden Markov Model have been proved to be a useful technique. The major contribution of the paper is our approach based on multi-order Markov chains, which reveals that the combination of mixed-order Markov chains would bring considerably interesting and substantial improvement over any single-order one with fairly reasonable cost. Utilizing not only the multiple order property, this approach effectively suits the application of anomaly detection in addition to its first practice in rainfall modeling. In our practice, the relative ranking positions between probabilities from multi-order models serve as a new effective indicator for anomalies, which refers to our finding that the ascending order suggests normal, while the descending one exhibits anomalous. Our approach differs from a recent model, which exploits a mixture of Markov chains by incorporating n-gram transitions to model the normal behavior of user's HTTP requests rather than system calls in underlying servers [9].

Requirement analysis

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements.

Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). How a system implements functional requirements is detailed in the system design. In some cases, a requirements analyst generates use cases after gathering and validating a set of functional requirements.

Non-Functional Requirements

Requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements, “and "non-behavioral requirements. Qualities, that is, non-functional requirements, can be divided into two main categories:
1. Execution qualities, such as security and usability, which are observable at run time [11].
2. Evolution qualities, such as testability, maintainability, extensibility, and scalability, which are embodied in the static structure of the software system.

Hardware Requirements

The following sub-sections discuss the various aspects of hardware requirements. Hardware Requirements For Present Project:
• System : Pentium IV 2.4 GHz.
• Hard Disk : 40 GB.
• Floppy Drive : 1.44 Mb.
• Monitor : 15 VGA Colour.
• Mouse : Logitech.
• RAM : 256 Mb

Software Requirements

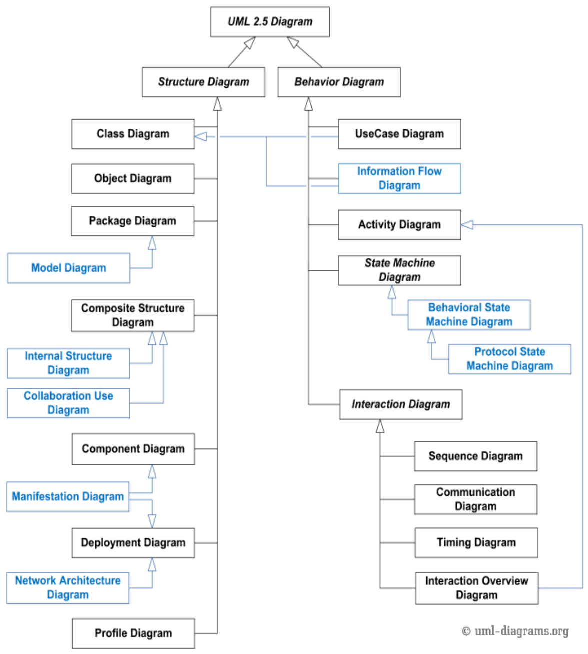
Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed [2].
Software Requirements for Present Project:
• Operating system : - Windows 7.

ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

• Front End : - JAVA or JSP
• Database : - SQL SERVER 2008
• Tools :- Eclipse IDE
SYSTEM DESIGN
Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. he UML has become the standard language used in Object-oriented analysis and design [citation needed]. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations [4].

Unified Modeling Language (UML)

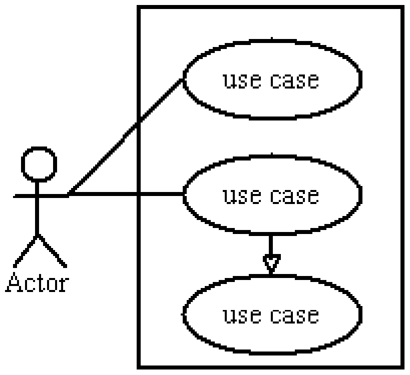
The Unified Modeling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram including elements such as:
• Any activity
• Individual component of the system
• And how they can interact with the other components
• How the system will run
• How entities interact with others (components and interfaces)



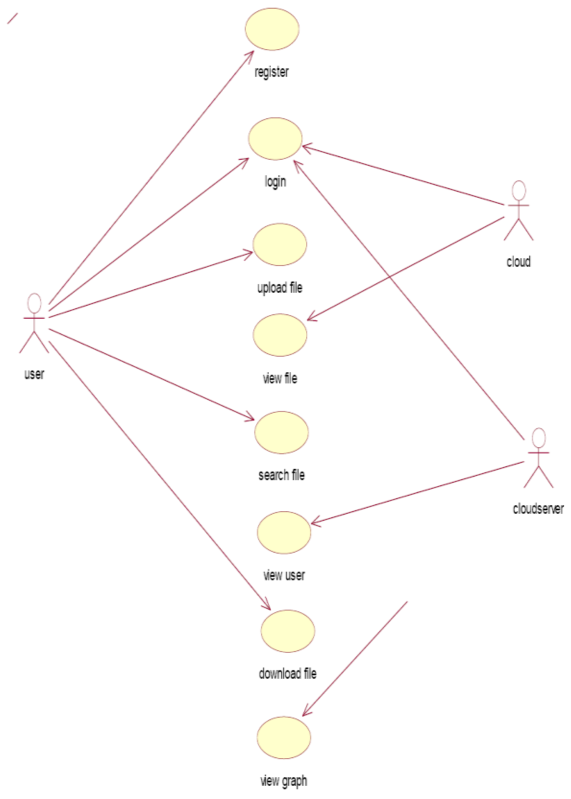
Classification of UML Diagrams

Use case Diagram

To model a system the most important aspect is to capture the dynamic behavior. To clarify a bit in detail, dynamic behavior means the behavior of the system when it is running /operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior [6].



Basic Use Case Diagram



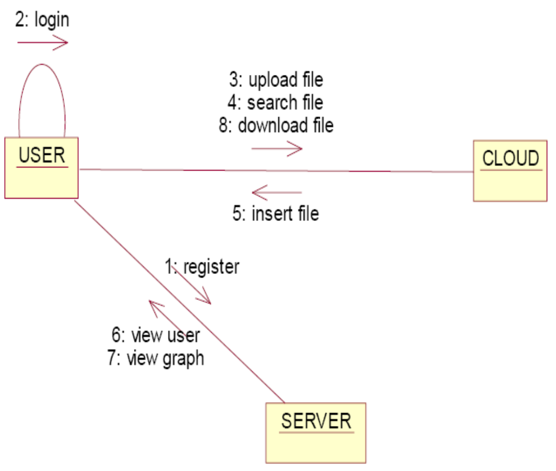
Project Use Case Diagram

ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

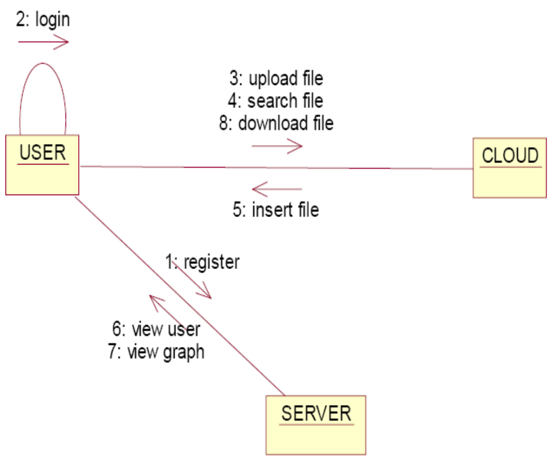
A Peer Reviewed Open Access International Journal

www.ijracse.com

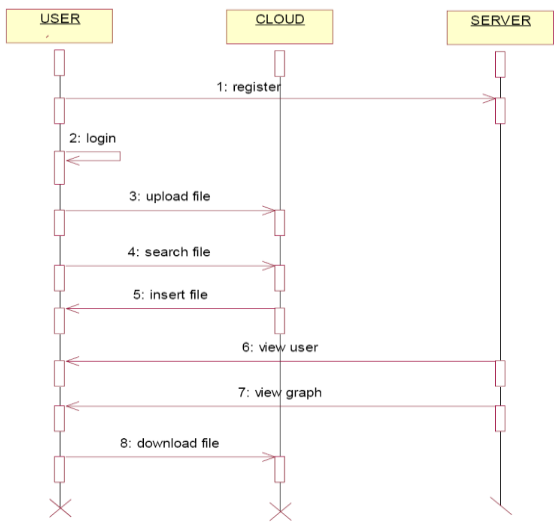


Collaboration Diagram

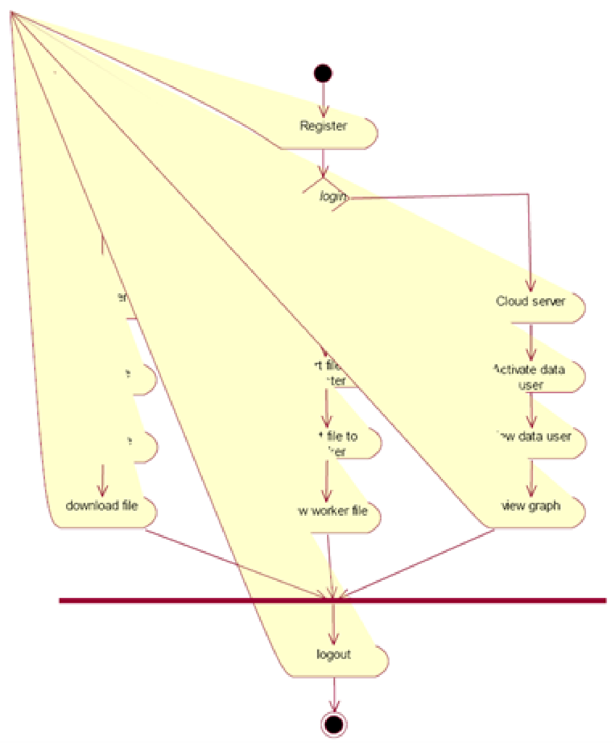
Class Diagram

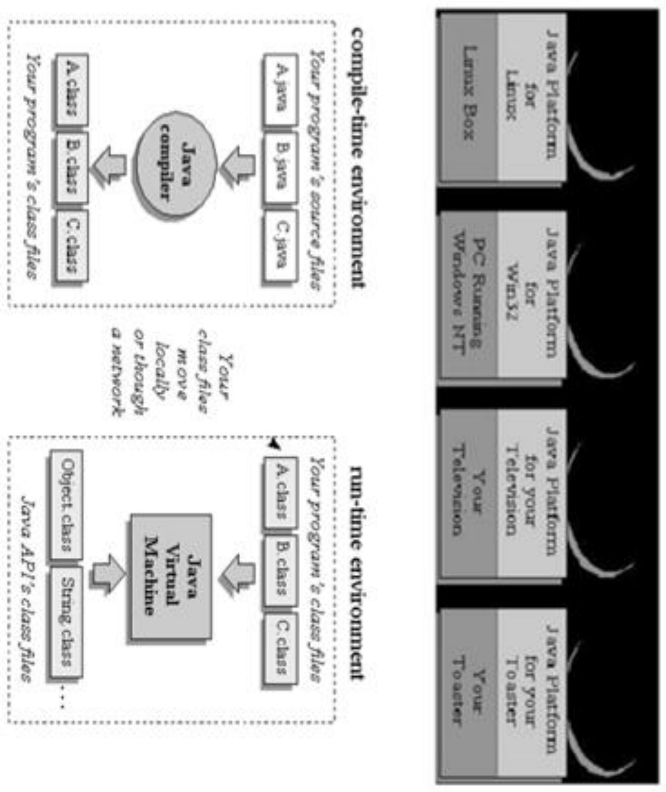


Sequence Diagram



Activity Diagram





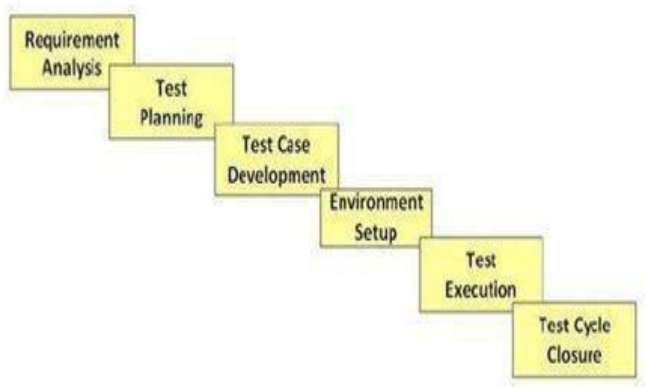
Java Programming Environment

ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

TESTING

Software testing can also be stated as the process of validating and verifying that a software program/application/product:
1. meets the business and technical requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.
Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted [8].

Software Test Life Cycle



Test Planning

This phase is also called Test Strategy phase. Typically, in this stage, a Senior QA manager will determine effort and cost estimates for the project and would prepare andfinalize the Test Plan
Activities:
- Preparation of test plan/strategy document for various types of Testing
Test tool selection.
- Testing Test tool selection

Test effort estimation

- Resource planning and determining roles and responsibilities.
- Training requirement

Test Environment Setup

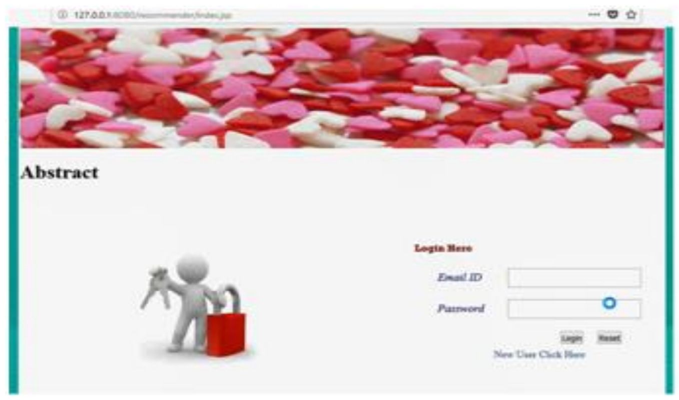
Test environment decides the software and hardware conditions under which a work product is tested. Test environment set-up is one of the critical aspects of testing process and can be done in parallel with Test Case Development Stage. Test team may not be involved in this activity if the customer team provides the test environment in which case the test team is required to do a readiness check (smoke testing) of the given environment.
Activities:
- Understand the required architecture, environment set-up and hardware and software requirement list for the Test Environment.
- Setup test Environment and test data
- Perform smoke test on the build

Test Execution

During this phase test team will carry out the testing based on the test plans and the test cases prepared. Bugs will be reported back to the development team for correction and retesting will be performed.
Activities:
- Execute tests as per plan
- Document test results, and log defects for failed cases
Map defects to test cases in RTM [10]
- Retest the defect fixes
- Track the defects to closure

RESULTS

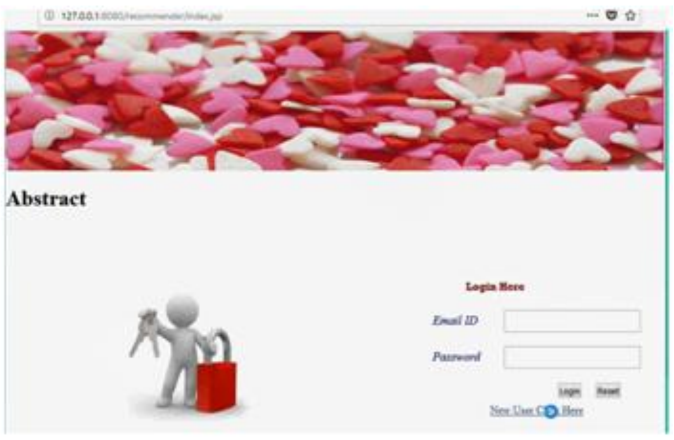
Login Portal Contains
- Abstract

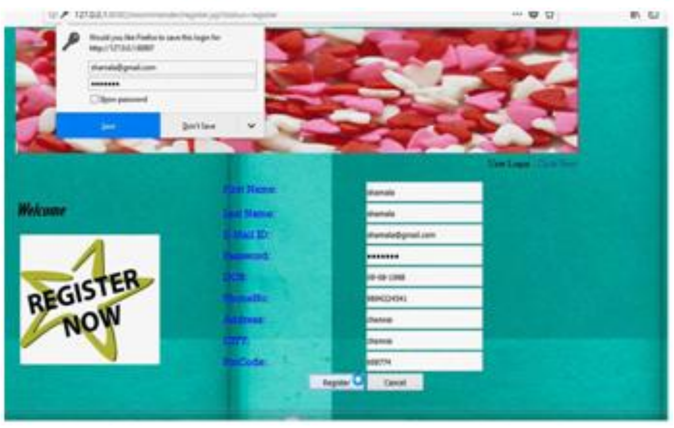


Login Portal

Creating the New User:

Click on the “New User Click Here”





Updating/Searching with New User

After Authorizing, now we can log into the User Account

New users can see below portals:

- Home
- Search Movies
- Recommender
- Improvement
- Logout

CONCLUSION

We proposed a multi-order Markov chain-based anomaly detection framework. Bymonitoring the relative relations between results from the different-order models, weprovide a new effective indicator of anomalies. In general, due to the regular andperiodical behaviors of cloud server systems, if the probability of test set given the lowerordermodel exceeds that given the higher-order one, it is implied that unusual eventsmight have occurred in the system and further attentions or actions would be necessary.Besides, combining multi-dimensional inter-related sequences as a multivariate one into asingle model would be another feasible approach to improve the sensitivity of detection.As shown previously, the return value series can be a useful complement to the systemcall series used the traditional practice.

REFERENCES

[1]. Denning, D.E., “An intrusion-detection model,” IEEE Transactions on Software Engineering. pp. 222 - 232, Feb. 1987

[2]. X. D. Hoang, J. Hu, and P. Bertok, “A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference,” Journal of Network and Computer Applications, vol. 32, pp. 1219-1228, 2009.

[3]. H. T. Elshoush and I. M. Osman, “Alert correlation in collaborative intelligent intrusion detection systemsłA survey,” Applied Soft Computing, vol. 11, pp. 4349-4365, 2011.

[4]. P. G. Bringas and Y. K. Penya, “Next-Generation Misuse and Anomaly Prevention System Enterprise Information Systems,” vol.19, J. Filipe and J. Cordeiro, Eds., ed: Springer Berlin Heidelberg, pp. 117-129, 2009.

[5]. S. Saravanakumar, AmruthKumar.A, Anandaraj.s, s.Gowtham,“Algorithms Based on Artificial Neural Networks for Intrusion etection in Heavy Traffic Computer Networks,” Proc. of Int’l Conf. on Advancements in Information Technology, pp.6-23, 2011

[6]. C.V. Raman, AtulNegi, “A Hybrid Method to Intrusion Detection Systems Using HMM,” Distributed

Computing and Internet Technology. pp. 389 - 396, 2005

[7]. Yihua Liao, V.RaoVemuri, “Use of K-nearest Neighbor Classifier for Intrusion Detection,” Computers & Security. 21(5): 439-448, 2002

[8]. Nong Ye, “A Markov Chain Model of Temporal Behavior for Anomaly Detection,” Workshop on Information Assurance and Security. West Point, NY, June 2000.

[9]. Nong Ye, Xiangyang Li, Qiang Chen, Syed MasumEmran, MingmingXu, “Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans. 31(4):266-274, July 2001

[10]. Nong Ye, Yebin Zhang, Connie M. Borror, “Robustness of the Markov-Chain Model for Cyber-Attack Detection,” IEEE Transactions on Realiability. 53(1):116-123, March 2004

[11]. Robert Gwadera, Mikhail Atallah, “Markov Models for Identification of Significant Episodes,” Proceedings of the 5th International Conference on Data Mining. 200