Volume 3, Issue 8, August 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Mining Framework for Evolving User Profiles Involuntarily

**Sunitha Maram**
*Scholar/Department of Computer Science & Engineering, JNTUA/Madanapalle Institute of Technology & Science, Angallu,Madanapalle-517325,AP,India*

**Dr.E.Madhusudhana Reddy**
*Professor/Department of Computer Science & Engineering, JNTUA/Madanapalle Institute of Technology & Science,Angallu,Madanapalle-517325,AP,India*

***Abstract:*** ***Data mining is one the analytical tools for analyzing information. It allow user to analyze data from many different scope, categorize it, and summarize the relationships. Theoretically, data mining is practice of finding correlation or pattern among dozens of fields in large relational databases. This paper can be defined as the description of the user interests, characteristics, behaviour and as well as preference. User profiling is process of gathering, organize, and interpret the user profile information. Significant work has been carried out for profiling users, In general, the user profiles do not change according to the environment and goals of new users. Here mining the user information at registration and detecting the user behaviours based upon user profiles and satisfying user needs instantly. Then detect masquerades and secures user profiling.***

***Keywords— EVABCD,ABCD,EPLIB,usermodeling .***

## I. Introduction

Personal information agents have emerged to help users to manage with increasing amount of information available on Internet. Agents are assistant to perform several information related tasks such as finding filtering and monitoring relevant information. In order to provide personalized assistance personal agents rely on representations of user information interests and preference contained in user profiles. Information agents personalizing information related tasks based on the content of documents and knowledge about a user interests propose users an alternative to cope with the huge amount of information available on Web. For this cause they have mainly applied to several tasks such as searching, filtering and summarizing information through user or community of users. The efficiency of agents depends mostly on profile completeness and accuracy and consequently user profiling has become a key area in the development of information agents. Knowledge about computer users is very beneficial for assisting them, predicting their upcoming actions or detecting masquerades. In this paper, an approach for creating as well as recognize automatically the behaviour of user profiles from the commands. Specifically, computer user behaviour is represented as sequence of UNIX commands. This series is changed into a allocation of relevant subsequences in order to find out a profile that defines its behaviour. In that case, statistical methods are used for recognizing a user from the commands. For identifying a genuine UNIX user, by using different sources of UNIX command data.

### *A. Agent Behaviour Classifier Based On Distributions Of Relevant Subsequence's Of Commands*

ABCD for profiling and classifying computer users from a UNIX command prompt. The sequence of commands typed by user is segmented and stored in a tree structure, and the relevant subsequences will be evaluated by using a frequency-based method.

Although ABCD can be applied for creating and recognizing any behaviour profile represented by a sequence of commands, this research is focused on creating computer user profiles from a command-line interface. Specifically, ABCD is detailed using the UNIX commands environment. ABCD, as other behavior modelling methods, uses a library in which all the different user profiles recognized are stored. Then, a matching of the sequence to classify with the Profile Library is done. Thus, ABCD is divided into two stages:

1) *Construction of the User Behaviour Profiles:* In this phase, the sequences of commands typed by different UNIX users are analyzed and the corresponding profiles are created and stored in the Profile-Library.
2) *User Set*: The goal of this phase is to classify a *new* sequence of commands typed by a user into one of the profiles created in the previous phase.

It can be defined as the description of the user interests, characteristics, behaviours, and preferences. User profiling is to gather, organize, and interpret the user profile information. Significant work has to be carry out for profile users, but most of the user profiles will not modify according to the environment and new goal of the user. Adaptive approach for creating profiles and recognizes computer users. This approach Evolving Agent behaviour Classification based on Distributions of relevant events (EVABCD) and demonstrates the behaviour of an agent (computer user) as an adaptive allocation of relevant atomic behaviours. Once this representation has been evolve, EVABCD presents an evolving method for updating and evolving the user profiles and classify user. The approach presented is generalizable to all kinds of user behaviours represented by a sequence of events.

*© 2013, IJARCSSE All Rights Reserved* 

The UNIX operating system environment is used in this research for explaining and evaluating EVABCD.User behaviour is represented in this case by a sequence of UNIX commands typed by a computer user in a command prompt. Earlier research studies in this environment [3], [4] focus on detecting masquerades (individuals who impersonate other users on computer networks and systems) from sequences of UNIX commands. But, EVABCD create growing user profiles and classify new users into one of the earlier created profile. EVABCD in the UNIX environment can be divided into two phases:

- Creating and updating user profiles from the commands the users typed in a UNIX shell.
- Classify new sequence of commands into the predefined profiles.

Because we use an evolving classifier, it will be constantly learning and adapting the existing classifier structure to accommodate the newly observed up-coming behaviours. Once user is classified, relevant actions can be done.

The creation of a user profile from a sequence of UNIX commands should consider the consecutive order of the commands typed by the user and influence of user past experience. This feature motivates the idea of auto-mated sequence learning for computer user behaviour classification; if agent does not know the features that influence the behaviour of a user, Sequence of past actions to incorporate some of the historical context of the user. yet it is difficult, to build a classifier that will have a full description of all possible behaviours of the user, because these behaviours evolve with time, they are dynamic and new patterns may emerge as well as an old habit may be forgotten or stopped to be used. The descriptions of a particular behaviour itself may also evolve, so user assumes that each behaviour is described by one or more fuzzy rules. A conventional system does not capture the new patterns (behaviours) that could appear in the data stream once the classifier is build. The information produced by a user is often quite large. So agent need to manage with large amounts of data and process this information, because storing the complete data and analyze it in an offline mode would be impossible. In order to take into account these aspects, agent uses an evolving fuzzy-rule-based system that allows for the user behaviours to be dynamic and evolve.

The EVABCD approach for automatic clustering,classifier design, and categorization of the behaviour profiles of users. Original evolving user behaviour classifier is based on Evolving Fuzzy Systems and it takes into account the fact that the behaviour of any user is not fixed, but is rather changing. This approach can be applied to every behaviour represented by a sequence of events specified by command prompt (UNIX commands) environment. In order to classify an observed behaviour, our approach, as many other agent modelling methods [10], creates a library which contains the different expected behaviours. In this scenario, library is not a prefixed, but it is evolving, learning from the user behaviour and, it starts to be filled in "from scratch" by assigning temporarily to the library the first observed user as a model. This model is known as Evolving-Profile-Library (Eli), is continuously changing, evolving influenced by the changing user behaviours observed in the environment.

Thus, the proposed approach includes at each step the following two main actions: Creating and evolving the classifier. This action involves in itself two sub actions:

- Creating the user behaviour profiles. This sub action analyses the sequences of commands typed by different UNIX users online and creates the related profiles.
- Evolving the classifier. This sub action includes online learning and update of the classifier, including the potential of each behaviour to be a model, stored EPLIB.
- User classification. The user profiles created in the previous action are associated with one of the prototype from the EPLIB, and classify into classes created by the model.

*B. Construction Of The User Behavior Profile*

In order to construct a user behaviour profile in online mode from data stream, extract an ordered sequence of acknowledged events. Commands are inherently in order, it is considered in modelling process. According to this aspect and based on the work done in [2], in order to get the most representative set of subsequence's from a sequence, we propose the use of a tree data structure [11]. This structure was also used in [12] to classify different sequences and in [13], [14] to classify the behaviour patterns of a Robot Cup soccer simulation team.

Building of a user profile from a single sequence of commands is done by a three step process:

- Segmentation of the sequence of commands.
- Storage of the subsequence's in a tree.
- Creation of the user profile.

*C. Creation Of The User Profile*

Once the tree is created, the subsequence's that characterize the user profile and its relevance are calculated by traversing the tree. Frequency-based methods are used. In EVABCD, to evaluate the relevance of a subsequence, its relative frequency [15] will be calculated. Subsequence is defined as the ratio of the number of times the subsequence has been inserted into the tree and the total number of subsequence's of equal size inserted.In this step, the tree can be transformed into a set of subsequence's labelled by its value. In EVABCD, the set of subsequence is represented as a distribution of relevant subsequence's. Thus, we consider that user profiles are n-dimensional matrix, where each dimension of the matrix will represent a particular subsequence of commands. User behaviour profile, once been created, it is classified and update the Evolving-Profile-Library.

*D. Evolving Unix User Classifier*

A classifier is a mapping from the feature space to the class label space. Classifier, the feature space is defined by distributions of subsequence's of events; the class label space is represented by the most representative distributions.

*© 2013, IJARCSSE All Rights Reserved* 

Allocating class label space represents a specific behaviour which is one of the prototype of EPLIB. The prototype is not fixed and change taking place into account the new information collected online from the data stream this is what makes the classifier growing. The prototypes is not prefixed but it depends on the homogeneity of the behaviours.

*E. User Behavior Representation*

EVABCD receives observations in real time from the environment to analyses and observations are UNIX commands and they are converted into the corresponding distribution of subsequence's online. In order to classify UNIX user behaviour, these distributions must be represented in a data space. Each distribution is considered as a data vector that defines a point that can be represented in the data space.

The data space can represent these points consist of n dimensions; where n is the number of the different subsequence's that might be observed. It means that the agent should know all the different subsequence's of the environment. The value is unknown and the creation of data space from the beginning is not efficient. So, in EVABCD, the dimension of the data space also evolves; it is incrementally growing according to the different subsequence's that are represented in it.

*F. User Profile Acquisition*

To provide personalized assistance agents must have certain knowledge about the user and the application domain. An agent's success depends mainly on the available information about users and its ability to represent user interests. Three approaches have been developed to provide agents with this knowledge the user-programming the knowledge-engineering and the machine-learning approaches.

In the user-programming approach agents are explicitly programmed by users from scratch by means of scripts or rules for processing information related to a particular task. An example ,where a user can create an electronic mail sorting agent by defining a number of rules that process incoming mail messages and sort them into different folders. The problem is that it requires too much insight understanding and effort from the user to recognize the opportunity to employ an agent take the initiative to create it supply the agent with explicit knowledge and maintain the underlying rules or scripts over time. In the knowledge-engineering approach consists of constructing an agent with domain-specific knowledge of both the application and the user .In this case, the task of programming the agent lays on the knowledge engineer instead of the end user. Agent information is comparatively fixed and consequently hardly adaptable to different application domains or to perform personalized tasks, an agent designed to help users to solve problems in the UNIX operating system by means of a large knowledge database.

The third and prevalent alternative is to endow agents with machine-learning mechanisms in order to allow them to acquire the knowledge they need to assist users. In such cases the agent is given limited background knowledge and it learns appropriate behaviour from the user and from other agents. The learning approach has two advantages. First, less effort from both users and knowledge engineers is required in agent construction as profiles are automatically acquired. Second, the agent can easily adapt profiles to changes in user interests over time and becomes customized to individual preferences and habits.

Finally the knowledge gained can be also transferred to other agents in a community. Before applying machine-learning techniques to personal agents the following conditions have to be fulfilled the use of the application has to involve a substantial amount of repetitive behaviour considering the actions of one user or among users and this repetitive behaviour has to be potentially different for each user.

*G. Observation Of User behavior*

The knowledge about users that has to be modelled into their profiles can be either implicitly or explicitly acquired from users leading to a division between explicit and implicit user profiles.

An explicit user profile is elicited from a series of questions designed to acquire user interests and preference accurately. The advantage is transparency of agent behaviour as decisions can be easily deduced from the data provided. It requires a great deal of effort from users and are not always able to express their interests because they are sometimes still unknown. A shallow model of user preferences and interaction patterns can be obtained based upon a relatively short-term interaction with an agent monitoring user behaviour by continuously 'looking over the user shoulder' as the user is performing actions. A more accurate model capturing user interests and goals can be in turn attained after a long-term interaction. Implicit knowledge is a mechanism ,it has no impact on the user regular activities Unobtrusive monitoring of users allows agents to discover behavioural patterns that can be used to infer user interests preferences and habits In order to achieve this goal a number of heuristics are commonly employed to infer facts from existing data In the course of their activities users leave behind a trail of information that can be used to model their interests Some sources of information left by a user after browsing include the history of the user requests for current and past browsing sessions that is maintained by most browsers bookmarks giving a quick means for accessing a set of documents exemplifying user interests access logs where entries correspond to HTTP requests typically containing the client IP address time-stamp access method URL protocol status and file size personal homepages and material as well as their outgoing links.

Implicitly acquired knowledge requires some degree of interpretation to understand the user's real goals. e.g., if the user does not read a Web page suggested by an agent it can be considered uninteresting. It is an unreliable inherently error prone process which reduces the overall confidence in the resulting profiles. In explicit knowledge, information provides high confidence since it is provided by the users themselves and not acquired from indirect sources. Explicit acquisition requires the agent to interrupt the user to either provide feedback or instruct the agent. It is intrusive and provides no guarantee that the questions asked are answered truthfully or even that the questions asked are the right ones

*© 2013, IJARCSSE All Rights Reserved* 

to obtain the desired information. Masquerades in computer intrusion detection are people who use somebody else's computer account. Under the UNIX operating system users give commands. For example, a user might type more my file in order to read my file one screen at a time. In this example more is the command and my file is an argument to that command. Our data source is the UNIX acct auditing mechanism.

While an analysis using arguments as well as commands would be desirable, arguments of commands were not collected because of privacy concerns. Some commands recorded by the system are not explicitly typed by the user. For example, a shell file is a file that contains multiple commands, and running a shell file will cause all of its commands to be recorded. Other examples are so-called .profile files and make files. Names of executable programs are also interpreted as commands because they are recorded in the audit stream. Masquerading data are drawn from the data of masquerading users as follows: we determine the length of the masquerade and choose a masquerade and a start data block at random. The random choice is repeated if there are not enough contiguous masquerading data left or if the masquerading data were previously used.

## II. Conclusion

User behavior is not fixed but rather it deviations and evolves, The classifier is able to maintain up to date the created profiles using an Evolving Systems approach. EVABCD is one pass, non-iterative, recursive, and it has the possible to be used in an interactive mode. It is computationally very efficient and speed. EVABCD used to monitor, study, and detect abnormality based on a time-varying behavior of same users and to detect masqueraders.

## III. Future Enhancement

As a user behavior is not fixed but rather it changes and evolves, the proposed classifier is able to store up to date the created profiles using an Evolving Systems methodology. EVABCD is one pass, non iterative, recursive, and it has the possible to be used in an interactive mode; therefore, it is computationally very efficient and speed. EVABCD can also be used to monitor, study, and detect abnormalities based on a time-varying behavior of same users and to detect masqueraders.We can develop to other type of users such as e-services,digitical communications.Major approaches highlight the problem of masqueraders but fails to address them.Masqueraders are individuals who impersonate other users on computer networks and systems to perform unauthorized operations.So we propose to develop EVABCD equipping it with a behavioral IDS system that implements characteristic algorithms that considers the following aspects to alert an anolomy.

**References**

[1] Godoy.D and Amandi.A, "User Profiling in Personal Information Agents: A Survey," Knowledge Eng. Rev., vol. 20, no. 4, pp. 329 361, 2005.

[2] J.A. Iglesias, A. Ledezma, and Sanchis .A, "Creating User Profiles from a Command Line Interface: A Statistical Approach," Proc. Int'l Conf. User Modeling, Adaptation, and Personalization (UMAP), pp. 90 101, 2009.

[3] Schonlau .M, Dumouchel.W, W.H. Ju, Karr, and Theus, "Computer Intrusion: Detecting Masquerades," Statistical Science, vol. 16, pp. 58 74, 2001.

[4] R.A. Maxion and Townsend .T.N, "Masquerade Detection Using Truncated Command Lines," Proc. Int'l Conf. Dependable Systems and Networks (DSN), pp. 219 228, 2002.

[5] Alaniz Macedo .A, K.N. Truong, Camacho Guerrero , and Graca Pimentel, "Automatically Sharing Web Experiences through a Hyperdocument Recommender System," Proc. ACM Conf. Hypertext and Hypermedia (HYPERTEXT '03), pp. 48 56, 2003.

[6] Pepyne .D.L, J. Hu, and W. Gong, "User Profiling for Computer Security," Proc. Am. Control Conf., pp. 982 987, 2004.

[7] Godoy .D and A. Amandi, "User Profiling for Web Page Filtering," IEEE Internet Computing, vol. 9, no. 4, pp. 56 64, July/Aug. 2005.

[8] Anderson .J, Learning and Memory: An Integrated Approach. John and Sons, 1995.

[9] Y. Horman and G.A. Kaminka, "Removing Biases in Unsuper vised Learning of Sequential Patterns," Intelligent Data Analysis, vol. 11, no. 5, pp. 457 480, 2007.

[10] Riley .P and M.M. Veloso, "On Behavior Classification in Adversarial Environments," Proc. Int'l Symp. Distributed Autonomous Robotic Systems (DARS), pp. 371 380, 2000.

[11] E. Fredkin, "Trie Memory," Comm. ACM, vol. 3, no. 9, pp. 490 499,1960.

[12] Iglesias .J.A, A. Ledezma, and Sanchis .A, "Sequence Classification Using Statistical Pattern Recognition," Proc. Int'l Conf.Intelligent Data Analysis (IDA), pp. 207 218, 2007.

[13] Kaminka G.A, M. Fidanboylu, A. Chang, and M.M. Veloso,"Learning the Sequential Coordinated Behavior of Teams from Observations," Proc. RoboCup Symp., pp. 111 125, 2002.

[14] J.A. Iglesias, A. Ledezma, and Sanchis .A, "A Comparing Method of Two Team Behaviours in the Simulation Coach Competition,"Proc. Int'l Conf. Modeling Decisions for Artificial Intelligence (MDAI),pp. 117 128, 2006.

[15] Agrawal .R and Srikant .R, "Mining Sequential Patterns," Proc.Int'l Conf. Data Eng., pp. 3 14, 1995.

*© 2013, IJARCSSE All Rights Reserved* 