Dynamic Evolving Modeling for User Behavior Profiles Automatically in real time Environments

K Sameena[1] T Chenthur Selvi[2]
[1,2]M.Tech CSE Students, Vignan Institute of Technology & science, Hyderabad

Abstract: One of the major problems of real time networks and social is security. Due to the abundant users of computer where the user's cognition in computers fix predicting their future actions or detecting masqueraders. In this project we are trying to creating and recognizing automatically the behavior profile of a computer user is presented. We focused more on computer users work behavior and actions to be performed in the future. The Evolving Systems approach works on updates in the user profiles with for obtaining a powerful self-learning online scheme. Finally the Dynamic Evolving Modeling for User Behavior Profiles applied automatically in real time Environments in sequence of actions or events. It works on several real data streams.

1. INTRODUCTION

Globalization has made rapid change in the cyber technologies as a result users are getting increased day by day. Even though it is good sign in the technology enhancement then also the real time systems and many social networks are facing security problems more. Because of lack of knowledge in using computers. Knowledge about computer users is very beneficial for assisting them, predicting their future actions or detecting masqueraders. Day by day number of users are getting have been increasing by creating their profiles. The Unauthorized/ masqueraders concentrate more on computer users work behavior and sequences of steps to be performed by them. Recognizing the behavior of others in real time is a significant aspect of different human tasks in many different environments. When this process is carried out by software agents or robots, it is known as user modeling [1]. Most existing techniques for user recognition assume the availability of handcrafted user profiles, which encode the a-priori known behavioral repertoire of the observed user. However, the construction of effective user profiles is a difficult problem for different reasons: human behavior is often erratic, and sometimes humans behave differently because of a change in their goals. This last problem makes necessary that the user profiles we create evolve [1].

2. LETARATURE SURVEY

There exist several definitions for user profile [2]. It can be defined as the description of the user interests, characteristics, behaviors, and preferences. User profiling is the practice of gathering, organizing, and interpreting the user profile information. In recent years, significant work has been carried out for profiling users, but most of the user profiles do not change according to the environment and new goals of the user. An example of how to create these static profiles is proposed in a previous work [3].

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 the influence of his/her past experiences. This aspect motivates the idea of automated sequence learning for computer user behavior classification; if we do not know the features that influence the behavior of a user, we can consider a sequence of past actions to incorporate some of the historical context of the user. However, it is difficult, or in general, impossible, to build a classifier that will have a full description of all possible behaviors of the user, because these behaviors evolve with time, they are not static and new patterns may emerge as well as an old habit may be forgotten or stopped to be used. The descriptions of a particular behavior itself may also evolve, so we assume that each behavior is described by one or more fuzzy rules.

Different techniques have been used to find out relevant information related to the human behavior in many different areas. The literature in this field is vast; Macedo et al. [6] propose a system (WebMemex) that provides recommended information based on the captured history of navigation from a list of known users. Pepyne et al. [6] describe a method using queuing theory and logistic regression modeling methods for profiling computer users based on simple temporal aspects of their behavior. In this case, the goal is to create profiles for very specialized groups of users, who would be expected to use their computers in a very similar way. Gody and Amandi [8] present a technique to generate readable user profiles that accurately capture interests by observing their behavior on the web. There is a lot of work focusing on user profiling in a specific environment, but it is not clear that they can be transferred to other environments. However, the approach we propose in this paper can be used in any domain in which a user behavior can be represented as a sequence of actions or events. Because sequences play a crucial role in human skill learning and reasoning [9], the problem of user profile classification is examined as a problem of sequence classification.

Popular approaches to such learning include statistical analysis and frequencybased methods. Lane and Brodley [11] present an approach based on the basis of instance-based learning (IBL) techniques, and several techniques for reducing data storage requirements of the user profile. Although the proposed approach can be applied to any behavior represented by a sequence of events, we focus in this research in a command-line interface environment. Related to this environment, Schonlau et al. [4] investigate a number of statistical approaches for detecting masqueraders. Coull et al. [12] propose an effective algorithm that uses pairwise sequence alignment to characterize similarity between sequences of commands. Recently, Angelov and Zhou propose in [12] to use evolving fuzzy classifiers for this detection task.

3. RELATED WORK

Existing System:

Most existing techniques for user recognition assume the availability of handcrafted user profiles, which encode the a-priori known behavioral repertoire of the observed user. However, the construction of effective user profiles is a difficult problem for different reasons: human behavior is often erratic, and sometimes humans behave differently because of a change in their goals. This last problem makes necessary that the user profiles we create evolve. But it failed to explain when the most of the user profiles do not change according to the environment and new goals of the user.

Proposed System:

In this paper, we propose an adaptive approach for creating behavior profiles and recognizing computer users. We call this approach Evolving Agent behavior Classification based on Distributions of relevant events (EVABCD) and it is based on representing the observed behavior of an agent (computer user) as an adaptive distribution of her/his relevant atomic behaviors (events). Once the model has been created, EVABCD presents an evolving method for updating and evolving the user profiles and classifying an observed user. The approach we present is generalizable to all kinds of user behaviors represented by a sequence of events.

The advantages of proposed system are,

It can cope with huge amounts and data. Its evolving structure can capture sudden and abrupt changes in the stream of data. Its structure meaning is very clear, as we propose a rule-based classifier. It is non-iterative and single pass; therefore, it is computationally very efficient and fast. Its classifier structure is simple and interpretable.

A user behavior is represented in this case by a sequence of UNIX commands typed by a computer user in a command-line interface. Previous research studies in this environment [4], [5] focus on detecting masquerades (individuals who impersonate other users on computer networks and systems) from sequences of UNIX commands. However, EVABCD creates evolving user profiles and classifies new users into one of the previously created profiles. Thus, the goal of EVABCD in the UNIX environment can be divided into two phases:

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

Because we use an evolving classifier, it is constantly learning and adapting the existing classifier structure to accommodate the newly observed emerging behaviors.

In this paper, we used an adaptive approach for creating behavior profiles and recognizing computer users. We call this approach Evolving Agent behavior Classification based on Distributions of relevant events (EVABCD) and it is based on representing the observed behavior of an agent (computer user) as an adaptive distribution of her/his relevant atomic behaviors (events). Once the model has been created, EVABCD presents an evolving method for updating and evolving the user profiles and classifying an observed user. The approach we present is generalizable to all kinds of user behaviors represented by a sequence of events.

4. DYNAMIC EVOLVING MODELING

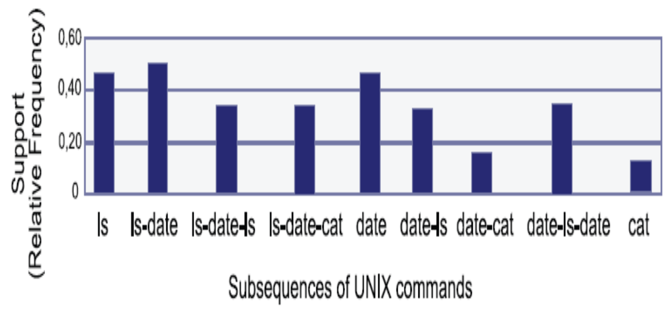
In order to construct a user behavior profile in online mode from a data stream, we need to extract an ordered sequence of recognized events; in this case, UNIX commands. These commands are inherently sequential, and this sequentiality is in the modeling process. According to this aspect and based on the work done in [3], in order to get the most representative set of subsequences from a sequence, we propose the use of a trie data structure . This structure was also used in to classify different sequences and in to classify the behavior patterns of a RoboCup soccer simulation team. The construction of a user profile from a single sequence of commands is done by a three step process:

1. Segmentation of the sequence of commands.
2. Storage of the subsequences in a trie.
3. Creation of the user profile.

These steps are detailed in the following three sections. For the sake of simplicity, let us consider the following sequence of commands as an example: {fls→ date → ls → date → catg}.

Creation of the User Profile

Once the trie is created, the subsequences that characterize the user profile and its relevance are calculated by traversing the trie. For this purpose, frequency-based methods are used. In particular, in EVABCD, to evaluate the relevance of a subsequence, its relative frequency or support [36] is calculated. In this case, the support of a subsequence is defined as the ratio of the number of times the subsequence has been inserted into the trie and the total number of subsequences of equal size inserted.

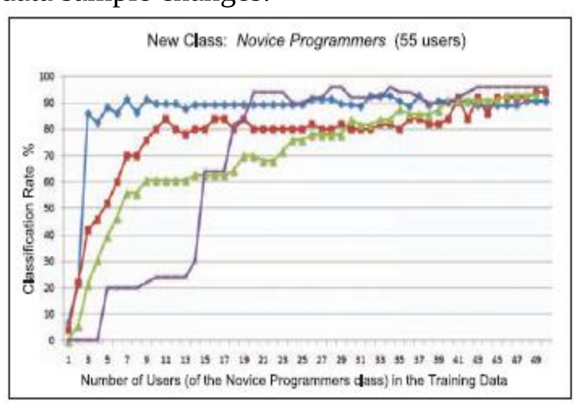


Source: Creating Evolving User Behavior Profiles Automatically @IEEE 2012

Structure of the EVABCD

Once the corresponding distribution has been created from the stream, it is processed by the classifier. The structure of this classifier includes

1. Classify the new sample in a class represented by a Prototype
2. Calculate the potential of the new data sample to be a prototype
3. Update all the prototypes considering the new data sample. It is done because the density of the data space surrounding certain data sample changes.



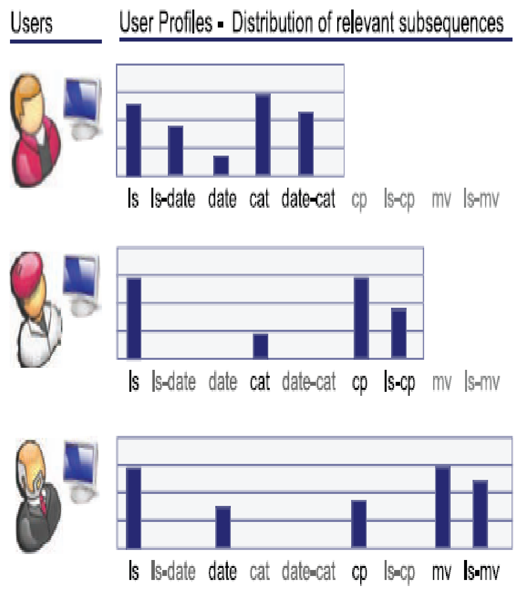


Fig 2. Distributions of subsequences of events in an evolving system approach Example [1].

User Behavior Representation

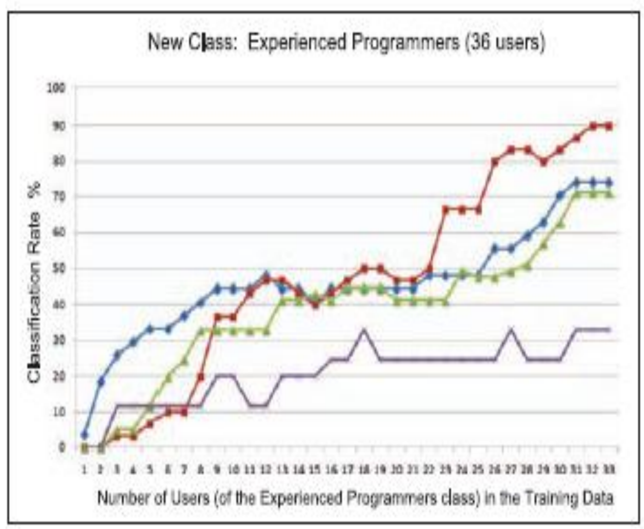
EVABCD receives observations in real time from the environment to analyze. In our case, these observations are UNIX commands and they are converted into the corresponding distribution of subsequences online. In order to classify a UNIX user behavior, these distributions must be represented in a data space. For this reason, each distribution is considered as a data vector that defines a point that can be represented in the data space. The data space in which we can represent these points should consist of n dimensions, where n is the number of the different subsequences that could be observed. It means that we should know all the different subsequences of the environment a priori. However, this value is unknown and the creation of this data space from the beginning is not efficient. For this reason, in EVABCD, the dimension of the data space also evolves; it is incrementally growing according to the different subsequences that are represented in it.

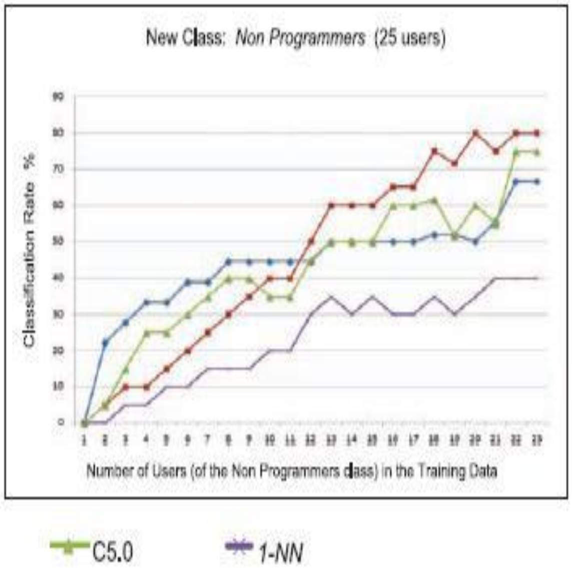
Experimental Setup and Results

In order to evaluate EVABCD in the UNIX environment, we use a data set with the UNIX commands typed by 168 real users and labeled in four different groups. Therefore, in these experiments, we use supervised learning. The explained process is applied for each of the four groups (classes) and one or more prototypes will be created for each group. EVABCD is applied in these experiments considering the data set as pseudo-online streams. However, only using data sets in an offline mode, the proposed classifier can be compared with other incremental and non incremental classifiers.

Group of users name	Sample Size	Total number of command lines
Novice Programmers	47	63.323
Exper.Programmers	34	61.906
Computer Scientist	52	125.608
Non Programmers	23	23.601
Total	**156**	**274.438**

Table 1. Total Number of Different Subsequences Obtained [1].





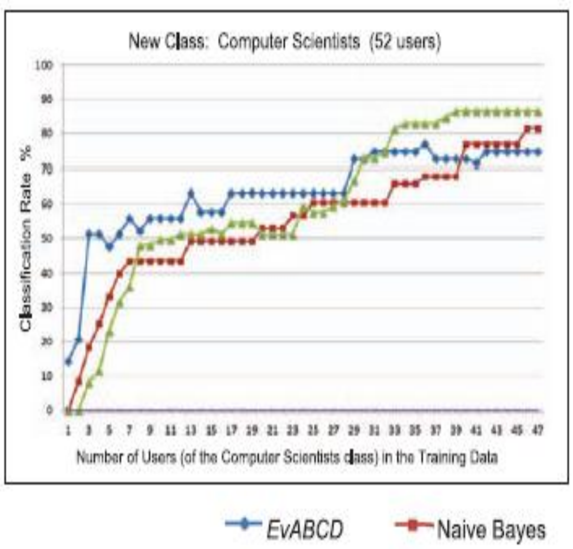


Fig 3.Evolution of the classification rate during online learning with a subset of UNIX users data set [1].

6. CONCLUSIONS

In this project, we used the concepts of Evolving Agent behavior Classification based on Distributions of relevant events, to model and classify user behaviors from a sequence of events. Then, a distribution of relevant subsequences is created. However, as a user behavior is not fixed but rather it changes and evolves, the proposed classifier is able to keep up to date the created profiles using an Evolving Systems approach by using with a data set of 76 real UNIX users demonstrates that, using an appropriate subsequence length, it can perform almost as well as other well-established offline classifiers in terms of correct classification on validation data. It is not addressed that, it can also be used to monitor, analyze, and detect abnormalities based on a time-varying behavior of same users and to detect masqueraders. It can also be applied to other type of users such as users of e-services, digital Communications. The scope of this project is, as a user behavior is not fixed but rather it changes and evolves.

5. REFERENCES

[1]. Creating Evolving User Behavior Profiles Automatically, © 2012 IEEE by Jose Antonio Iglesias, Member, IEEE Computer Society, Plamen Angelov, Senior Member, IEEE Computer Society, Agapito Ledezma, Member, IEEE Computer Society, and Araceli Sanchis, Member, IEEE Computer Society.

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

[3] J.A. Iglesias, A. Ledezma, and A. Sanchis, “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.

[4] A. Alaniz Macedo, K.N. Truong, J.A. Camacho Guerrero, and M. Graca Pimentel, “Automatically Sharing Web Experiences through a Hyperdocument Recommender System,” Proc. ACM Conf. Hypertext and

Hypermedia (HYPERTEXT ’03), pp. 48 56, 2003.

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

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

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

[8] 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.

[9] T. Lane and C.E. Brodley, “Temporal Sequence Learning and Data Reduction for Anomaly Detection,” Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 150 158, 1998.

[10] S.E. Coull, J.W. Branch, B.K. Szymanski, and E. Breimer, “Intrusion Detection: A Bioinformatics Approach,” Proc. Ann.Computer Security Applications Conf. (ACSAC), pp. 24 33, 2003.

[11] P. Angelov and X. Zhou, “Evolving Fuzzy Rule Based Classifiers from Data Streams,” IEEE Trans. Fuzzy Systems: Special Issue on Evolving Fuzzy Systems, vol. 16, no. 6, pp. 1462 1475, Dec. 2008.

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

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