An Adaptive Appropach Foe Creating Behaviour Profiles And Recognizing Computer Users

Mounika Reddy Mereddy (M.Tech)
CSE Dept, MRCET,

Devi Sudha, M.Tech,
Asst. Professor, CSE Dept,
MRCET, Hyderabad.

Abstract: — A detail of a persons who use the computer is plays a vital role in assisting them, foreseeing their future actions. Through this paper, an attempt for creating and recognizing the profile behavior of a person who uses the computer is presented by itself. In this case, a person who uses the computer behavior is shown as the series of the instructions while the person is using keypad during operations. This series is converted into a distribution of relevant subseries of instructions in order to puzzle out a profile that specifies its behavior and a user profile may not necessarily the same but rather it evolves/changes. We suggest an developing method to maintain day to day updating profiles which were created with the help of developing Systems approach. Through this paper, we mix or join the developing classifier with a tree-based user profiling to obtain a manually crafted self-learning networking. We further implemented the ability criteria of recursive formula a data point to become a nodal center with the utilization of cosine distance that was present in the Appendix. The way of approaching suggested through this paper can be applied to any consequence of dynamic/developing user behavior modeling where it can be shown as a series of operations. It several real data streams that will get evaluated.

Index Terms— Evolving fuzzy systems, fuzzy-rule-based (FRB) classifiers, user modeling.

I. Introduction

Recognizing the behavior of others in practical is a significant aspect of human tasks differs a lot among different environments. When this process is run by robo-sapiens, it is known as user modeling. The identification of customers will get benefits for giving them and broadcasting their further processing's. Well known existing techniques for user identification assume the availability of manually crafted customer profiles; those are encoded by particular user a-priori known behavioral repertoire. Often, the user profile which was effective is a critical consequence for various reasons like: erratic human behavior, sometimes humans act differently because of a change in their destinies. And because of this last consequence makes it necessary for the user profiles we create to evolve. Several definitions are present for user profile. Defined as need of the features, performance, interests and working. Customer information is the experiment of collecting, maintaining, and cloning the customer information. Now days, significant work has been run for profiling users, but most important point to be noted is that the user profiles are not updated according to the environment and new destinies of the user. An example of different ways to create these static profiles is suggested in a previous work. Through this paper, we suggest an approach that suits the best in creation of behavior profiles and recognizing persons who use the computers. We call this approach Developing Agent behavior Classification based on Segregations of similar events (EVABCD) and depends upon reflecting the noticed behavior of person like adaptive distribution of their atomic behaviors which are relevant. The model that has been created once, which an EVABCD presents and developing method for updating and developing the user profiles and also for segregating an observed user. The suggested approach which we present you is applicable to all kinds of user behaviors shown by a series of works. The UNIX operating system environment is made for evaluating EVABCD. A user nature is shown in this case by a series of UNIX instructions typed by a person who uses the computer in a command-line interface. Many case studies which have been done before on this environment focus on detecting masquerades from series of

UNIX instructions. However, EVABCD creates developing user profiles and classifies new users who are joining into already existing profiles previously. There, the destiny of EVABCD can be subdivided into two phases in the UNIX environment:

1. Updating the created user profiles from the instructions the users typed in a UNIX shell.

2. Segregateing a new series of instructions into the predefined profiles.

Because we use a constant learning and adapting the existing classifier structure which one among the developing classifiers. This classifier is to accommodate the modern emerging behaviors. A classified user can do the relevant action anyhow this isn't discussed in this paper.

II PROBLEM STATEMENT

- User profiling in an adoptive approach and recognizing a person who uses the computers is a consequence.
- Developing Agent Behavior Classification based on Segregations of relevant events (EVABCD) is the system suggested to address the profiling consequences.

Existing System: Statistical analysis and frequency-based methods. Instance-Based Learning (IBL).

Naïve Bayesian (NB), C4.5 and ID3 for network NIDS, doesn't suits for data stream mining. Because the instructions of users come as series or data stream, incremental learning algorithms can solve the data stream mining consequences such as user profiling and forecasting the behavior.

Suggested System

- Fuzzy-Rule-Base system is suggested to solve incremental classifiers. The approach is useful in real environments.
- Developing-Profile-Library (EPLib) is suggested for adopting changing user behaviors. Can be applied on real environments. Can be applied on huge amounts of data.
- Because it uses single pass, computational cost is very low.
- Creating and developing the classifier.
    - **a**. Creating the user behavior profiles.
    - **b**. Developing the classifier.
- User classification.

III SYSTEM DEVELOPMENT:

Trie Implementation:

- Trie is a data structure to store the sequence of

alphabets or words.

In the work it has to support words Inserting, updating the word frequency, path traversal, search for a word, Search for a sequence.

User Behavior Profiling

The module is responsible to update the current user profile. The operations are performed by calling the Trie functions

- Insert (Command)
- Search(Command)

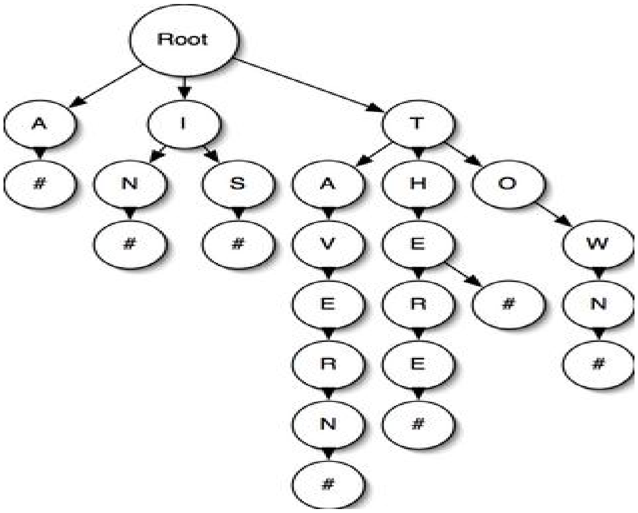

Fig : Example of trie

Segmentation of Sequence of words

The updated Trie consists of all the profiles of all the users. The paths of the trie represents the sequences of commands of all the users. The module identifies all the distinguished paths

Pre-Classification of User Profiles

The module lists all the pre-classified users profile list. Novice Programmer, Expert Programmer Computer Scientists, Non-Programmers.

User Profile Matching

The current users profiles has to identified as any of the known and pre-classified profiles.

The module is also responsible to identify the behavioral change.

The EVABCD should classify all the users.

IV RELATED WORK

By using a data set, we evaluate the EVABCD with the UNIX instructions which are typed by 168 real users and then they are labeled accordingly into four different groups. Therefore, supervised learning process is used in these experiments. To have detailed information regarding the various outcomes previously explained process that is the EVABCD process is applied for each class of the four groups (classes) divided considering the data set as pseudo-online streams and depending upon the necessity one or more prototypes will be created for each group. However, the suggested classifier can be compared with other incremental and no incremental classifiers only while using those data sets in an offline mode.

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 group 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 nonincremental classifiers.

Data Set

In these experiments, we use the command-line data collected using UNIX csh command interpreter. This data are identified in four target groups which represent a total of 168 male and female users with a wide cross section of computer experience and needs. Each major row corresponds to a training-set size and each such row is further subdivided into experiments with different subsequence lengths for segmenting the initial sequence . The columns show the average classification success using the proposed approach (EvABCD) and the other (incremental and nonincremental) classification algorithms. According to these data, SVM and Nonincremental Naive Bayes perform slightly better than the Incremental NB and C5.0 classifiers in terms of accuracy. The percentages of users correctly classified by EVABCD are higher to the results obtained by LVQ and lower than the percentages obtained by NB, C5.0, and SVM. Incremental and Nonincremental kNN classifiers perform much worse than the others. Note that the changes in the subsequence length do not modify the classification results obtained by AdaBoost. This is due to the fact that the classifier creates the same classification rule (weak hypotheses) although the subsequence length varies. In general, the difference between EVABCD and the algorithms NB and SVM is considerable for small subsequence lengths (two or three commands), but this difference decreases when this length is longer (five or six commands). These results show that using an appropriate subsequence length, the proposed classifier can compete well with offline approaches.

Nevertheless, the proposed environment needs a classifier able to process streaming data in online and in real time. Only the incremental classifiers satisfy this requirement, but unlike EVABCD, they assume a fixed structure. In spite of this, taking into account the incremental classifiers, it can be noticed that the difference in the results is not very significant; besides, EVABCD can cope with huge amounts of data because its structure is open and the rule-base evolves in terms of creating new prototypes as they occur automatically. In addition, the learning in EVABCD is performed in single pass and a significantly smaller memory is used. Spending too much time for training is clearly not adequate for this purpose.

In short, EVABCD needs an appropriate subsequence length to get a classification rate similar to the obtained by other classifiers which use different techniques. However, EVABCD does not need to store the entire data stream in the memory and disregards any sample after being used. EVABCD is one pass (each sample is proceeded once at the time of its arrival), while other offline algorithms require a batch set of training data in the memory and make many iterations. Thus, EVABCD is computationally more simple and efficient as it is recursive and one pass. Unlike other incremental classifiers, EVABCD does not assume a prefixed structure and it changes according to the samples obtained. In addition, as EVABCD uses a recursive expression for calculating the potential of a sample, it is also computationally very efficient. In fact, since the number of attributes is very large in the proposed environment and it changes frequently, EVABCD is the most suitable alternative. Finally, the EVABCD structure is simple and interpretable.

V. CONCLUSIONS

Through this paper, we suggest a generic approach, EVABCD, to model and segregate user behaviors from a series of events. The masked feature in this system is that the information gathered from the respective environment can be converted into a series of events. This series is segmented and stored in a tree and the relevant subseries are explored utilizing a frequency-based method. Relevant subseries distribution is developed. Moreover, customer behavior is not static so it keeps alterations and explores, the suggested to update the existing profiles with the help of Developing Systems approach. EVABCD non-iterative, single pass and recursive. Containing features to work to work in an interactive mode; therefore, it is theistically very efficient and fast. Addition to this, its structure is simple and interpretable. The suggested developing classifier is tested in an surrounding in which each user behavior is shown as a series of UNIX instructions. Although EVABCD has been developed to be made to work online, the experiments have been done using a batch data set in order to differentiate the performance to established (incremental and Non incremental) classifiers. The results of the tests done on a data set which consists 168 real UNIX users demonstrates that, by the use of an appropriate subseries length, EVABCD can perform to its maximum as well as other well-accomplished offline classifiers in terms of proper classification on validation data. However, considering that EVABCD has the ability of adapting itself to new data input within no time, and that this classifier can move along with huge data in a real environment which changes instantaneously, the suggested approach can be considered as the most appropriate alternative. Although, it is not addressed through this paper, EVABCD can also be made to work 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, etc.

REFERENCES

[1] D. Godoy and A. Amandi, "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 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.

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

[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.

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

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

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

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

[10] T. Lane and C.E. Brodley, "Temporal Series Learning and Data Reduction for Anomaly Detection," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 150-158, 1998.

First Author: ***Mounika Reddy*** obtained her Bachelor's degree in Computer Science and Engineering from Dr.Paul Raj Engineering College, Bhadrachalam. Currently she is doing her M.Tech in Computer Science and Engineering in MRCET, affiliated to JNTUH. Her specializations include network security, Web security, Intrusion detection systems and Data Mining.

Second Author: ***Devi Sudha*** obtained a Bachelor's degree in Navalar Nedunchezhiyan Engineering College, Tamil Nadu, in the year 2005. She is a Postgraduate in NIT, Trichy, Tamilnadu. Currently she has been working as an Assistant Professor in MRCET; she has total 6years of Teaching Experience. Guided many B.Tech and M.Tech students for their Project Work. Her specializations include network security, Web security, Intrusion detection systems and Data Mining.