DESIGN AND IMPLEMENTATION AN IDENTIFICATION SYSTEM BASED ON TYPING RHYTHM ON KEYBOARD

Navid Samimi Behbahan*

Zohreh Musavinasab*

Abstract: *This paper proposes a biometric authentication system which use password based and behavioral traits (typing behaviors) authentication technology Analysis of typing rhythms to discriminate among users has been proposed for detecting impostors (i.e., both insiders and external attackers). Since many anomaly-detection algorithms have been proposed for this task, it is natural to ask which are the top performers (e.g., to identify promising research directions). Unfortunately, we cannot conduct a sound comparison of detectors using the results in the literature because evaluation conditions are inconsistent across studies. In this study, fuzzy logic has been proposed. For optimizing of fuzzy rules, Genetic Algorithm is used.*

*Department of Computer Engineering, Omidiyeh Branch, Islamic Azad University, Omidiyeh, Iran

1- INTRODUCTION

Biometric systems are a group of technologies and techniques that can be used for identification. Biometric technology has many applications. The primary aim of technology, is providing of access control systems and security protocols to protect personal or corporate assets.

When biometrics is used to identify individuals, a sample of a biometric feature is extracted and person identification is carried out based on this model. Since computer keyboard is created as the primary interface between humans and computers has been introduced. In this study habits of typing will be used as biometric features to identify a person. It is proved that habits of typing such as signature is a behavioral biometric feature. Using of typing habits can bring very good results as a method of authentication and it can be as complement of some methods including password. Also in the LANs using of some type of identification methods such as typing habits can be very effective. Identification on how typing is one of the newest methods and this method analyzes, mode of stroke to the keyboard as it sounds from its name. To do this, the user is asked to enter a password or a specific text. System analysis intervals between press of keys and the data are stored as reference data. For this system, entering at least 8 characters, it is necessary but the entry of 12 characters or more is recommended. These characteristics can be from one to six fields such password, user name or e-mail. We combine this method with learning methods, to increase accuracy. That means that if the user frequently login to the system, the system will identify accurately her/him. The Figure 1 depicts a general biometric user authentication system.

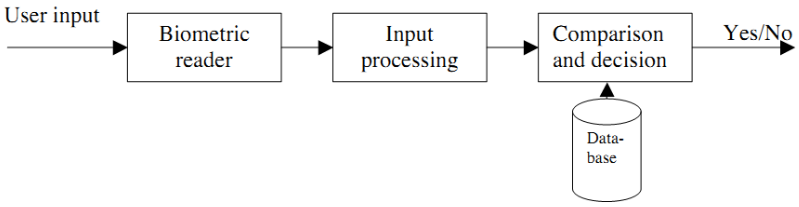

Figure 1: general biometric user authentication system

There are two possible approaches to achieve this, namely by measuring the time between consecutive keystrokes "Latency" or measuring the force applied on each keystroke. The pressure-based biometric authentication system (PBAS) has been designed to combine these two approaches so as to enhance computer security.

PBAS employs force sensors to measure the exact amount of force a user exerts while typing. Signal processing is then carried out to construct a waveform pattern for the password entered. In addition to the force, PBAS measures the actual timing traces "latency." The combination of both information "force pattern and latency" is used for the biometric analysis of the user.

As compared to conventional keystroke biometric authentication systems, PBAS has employed a new approach by constructing a waveform pattern for the keystroke pass-word. This pattern provides a more dynamic and consistent biometric characteristics of the user. It also eliminates the security threat posed by breaching the system through online network as the access to the system is only possible through the pressure sensor reinforced keyboard "biokeyboard".

Figure 2 shows PBAS block diagram. The operation of the system relies on constructing a users' database and then processing this information online through data classifiers. The database stores users' login names, passwords, and biometric patterns. Data classifiers are used to analyze and associate users with distinctive typing characteristic models.

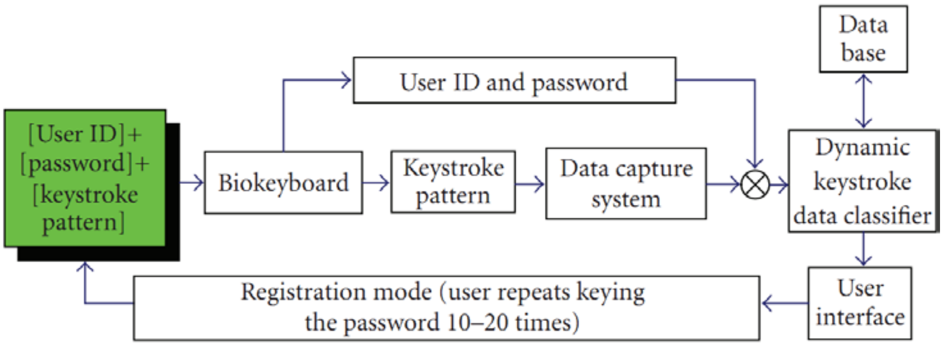
Data base
User ID and password
[User ID]+ [password]+ [keystroke pattern]
Biokeyboard
Keystroke pattern
Data capture system
Dynamic keystroke data classifier
User interface
Registration mode (user repeats keying the password 10–20 times)

Figure 2: PBAS block diagram

PBAS has been tested with combination of two classifiers, namely:

(1) autoregressive classifiers,

(2) latency classifiers.

These classifiers have been tested and the results obtained from the experimental setup have shown that these classifiers are very consistent and reliable [1].

2- RELATED WORKS

In 1986, the first keystroke recognition system was proposed by Garcia [2], who has successfully designed a personal identification apparatus by using keystroke recognition technique. And in Blender and others work, they found that if the system can achieve recording and analyzing user's input mode at the same time with user password identification, this dual protection mode will not only guarantees the user's actual space and data security, but also effectively prevent the invasion of hackers. On the other side, typing behavior recognition is not only used on desktop or laptop, Clarke and Furnell's [3] work is based on mobile devices, and they have noted that neural networks superior pattern classification method, but that mobile devices lack the computing power necessary to employ a neural network in situations where the processing is done on the device itself.

Overview the current researches, most of the studies are based on desktop and laptop keystroke dynamics, and others are based on numeric keyboard phone or Personal Digital Assistants (PDA) [4]. The typing behavior recognition system proposed in this paper is implemented on the latest smart phone platform, and it also uses multi-level authentication mechanism which can achieves the balance between security and usability.

3- APPLICATIONS

The first suggested use of keystroke characteristics for identification appeared in 1975, but observations about the uniqueness of an individual's typing characteristics stretch as far back as the end of the 19th century. Telegraph operators at the time could often identify each other by listening to the rhythm of their Morse code keying pat-terns. Let's look at some of the pertinent and interesting ways in which keystroke dynamics can be applied.

4- DATA SET

The data set that is used in this study is collected by Kevin Killourhy and Roy Maxion [5]. The data consist of keystroke-timing information from 51 subjects (typists), each typing a password (.tie5Roanl) 400 times. Whenever the subject presses or releases a key, the software application records the event (i.e., keydown or keyup), the name of the key involved, and a timestamp for the moment at which the keystroke event occurred. An external reference clock was used to generate highly accurate timestamps. The reference clock was demonstrated to be accurate to within ±200 microseconds (by using a function generator to simulate key presses at fixed intervals). They recruited 51 subjects (typists) from within a university community; all subjects fully completed the study—we did not drop any subjects. All subjects typed the same password, and each subject typed the password 400 times over 8 sessions (50 repetitions per session). They waited at least one day between sessions, to capture some of the day-to-day variation of each subject's typing. The password

(.tie5Roanl) was chosen to be representative of a strong 10-character password. The raw records of all the subjects' keystrokes and timestamps were analyzed to create a password-timing table. The password-timing table encodes the timing features for each of the 400 passwords that each subject typed.

5- OUR APPROACH

Our method is based on fuzzy logic. As we know, the data set is considered, consisting of 33 features. So, because of the large number of features and samples, using techniques such as fuzzy learning is a necessity. For reduce of features number, the PCA Algorithm is selected in our study. Fewer features, leading to shorter training time and is more accurate in detecting.

In this paper, a GA-based method to evolve a fuzzy expert system is discussed. It not only can evolve the rule set\ (including the optimal number of rules inside the rule set), tune the membership functions, and evolve the membership function types, but also scales well and is, therefore, useful for large complex problems. In addition, a fuzzy expert system is designed from our experience and knowledge and is used to adapt the genetic parameters of the GA.

5-1- Genetic Algorithm

GA paradigms do not require information that is auxiliary or related to the problem such as function derivatives, while many hill-climbing search paradigms, for example, require the calculation of derivatives in order to successfully explore the local maximum or minimum. So GA's can be applied to wider areas, especially those difficult for traditional hill-climbing methods. A typical series of operations carried out when implementing a GA paradigm is:

1) Initialize the population;

2) Calculate fitness for each chromosome in population;

3) Reproduce selected chromosomes to form a new population;

4) perform crossover and mutation on the population;

5) Loop to step 2 until some condition is met.

Initialization of the population is commonly done by seeding the population with random values. The fitness value is proportional to th e performance measurement of the function being optimized. The calculation of fitness values is conceptually simple. It can, however, be quite complex to implement in a way that optimizes the efficiency of the GA's search of the problem space. It is this fitness that guides the search of the problem space.

After fitness calculation, the next step is reproduction. Reproduction comprises forming a new population, usually with the same total number of chromosomes, by selecting from members of the current population using a stochastic process that is weighted by each of their fitness values. The higher the fitness, the more likely it is that the chromosome will be selected for the new generation. One commonly used way is a ''roulette wheel'' procedure that assigns a portion of a roulette wheel to each population member where the size of the portion is proportional to the fitness value. This procedure is often combined with the elitist strategy, which ensures that the chromosome with the highest fitness is always copied into the next generation.

The next operation is called crossover. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms. Crossover is the process of exchanging portions of the strings of two ''parent'' chromosomes. An overall probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur. This probability is often in the range of 0.65–0.80. The final operation in the typical GA procedure is mutation. Mutation consists of changing an element's value at random, often with a constant probability for each element in the population. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA. However, values of between 0.001 and 0.01 are not unusual for mutation probability.

In the example simulation in this paper, the ''roulette wheel'' procedure with the elitist strategy is used for reproduction, where the portions of the roulette wheel assigned to population members are proportional to the shifted fitness values. The original fitness values are linearly shifted with the minimal fitness mapping to 0.1. The crossover operator used is two-point crossover with a default crossover probability of 0.75. The mutation operator used in this paper depends on our chromosome representation and will be explained later. Note that in our evolutionary fuzzy system described in Section IV, fuzzy rules can be used to adapt crossover probability and mutation rate.

5-2- How usage of fuzzy logic in the proposed method

Let us consider a data set X = {x^1, x^2, …, x^n} ⊆R^s, and the predetermined classification matrix, denoted by A'. This matrix, produced by human experts, shows an a priori split of the *n* data items in *k* different classes. In such a case, the matrix A' is a Boolean matrix indicating the membership of a data item to one of the k classes. One of the major issue human experts have is that they think in crisp terms. This means that the a-priori classes are defined in crisp terms. This is not a realistic decision, since in almost all real situations data is of a fuzzy nature. The given data classes most certainly have data items close to the central locations, but they have as well distant data items, also called outliers. As such, a preprocessing step must be done: for the crisp a-priory classes, suitable fuzzy regression sets will be determined. For each original class, a fuzzy regression with point prototypes is applied and fuzzy membership degrees are thus determined. We recall the main details here. The optimal fuzzy set A that best describes the given crisp set, and the associated point prototype L ∈R^s, are determined by minimizing the following fuzzy objective function:

$$J(A,L) = \sum_{j=1}^{n} A(x^j)^m \left\|x^j - L\right\|^2 + \sum_{j=1}^{n} \left(1 - A(x^j)\right)^m \left(\frac{\alpha}{1-\alpha}\right)^{m-1}$$

Where α is a positive subunit value set a-priory, identifying the fuzzy membership degree of the farthest outlier and m>1 is the fuzziness index, set a-priori. The algorithm used

To solve this problem has been called Fuzzy Regression and iterates by computing the prototype L that minimizes the function J(A,·)and by computing the fuzzy set A that minimizes the function J(·,L). As an improvement to this method, in order to ensure the independence of scale, we usually work with the relative dissimilarity when determining the fuzzy set above, i.e.

$$A(x^j) = \frac{\frac{\alpha}{1-\alpha}}{\frac{\alpha}{1-\alpha} + \left(\frac{\|x^j - L\|}{\max\|x^j - L\|}\right)^{\frac{2}{m-1}}}$$

Complete details of this fuzzy regression procedure and other variants thereof are given in . Of course, this means that the result will be a sub-partition, i.e. the sum of membership degrees of a point to all classes is less than one. But, on the other side, this preprocessing step allows us to show light on the input data and the quality data items from each original cluster, as it has been a-priori proposed. As opposed to this, other methods use either an unsupervised clustering scheme here (which we find it is in principle un-appropriate), or use different mechanisms to set the membership degrees without any functional optimization.

The Fuzzy Discriminant Analysis problem is defined as follows: let X = {x^1, x^2, ..., x^n} ⊂R^s be a finite set of characteristic vectors, where **n** is the number of items and **s** is the number of the original variables (predictors), $x^j = [x_1^j, x_2^j, \dots, x_s^j]^T$ and let ***Ai*** (with *i* = 1, ..., *k*) be fuzzy sets on **X**, corresponding to the k a-priori sets composing the partition substructure of the given data set. A new vector (or characteristic) c is to be determined, that maximizes the fuzzy between-class variance of the projected data items, and minimizes the fuzzy within-class variance of the projected data items.

Considering this new characteristic defined as **c = Xu**, the fuzzy between-group variance **B** and fuzzy within-group variance **W**, are defined as:

$$\mathbf{W} = \frac{1}{n-k}\sum_{i=1}^{k}\left(\sum_{j=1}^{n} A_i\left(x^j\right)^m \left(x^j - L^i\right)^T \left(x^j - L^i\right)\right),$$

$$\mathbf{B} = \frac{1}{k-1}\sum_{i=1}^{k}\left(\sum_{j=1}^{n} A_i\left(x^j\right)\right)^m \left(L^i - L\right)^T \left(L^i - L\right),$$

Where the class means L^i are determined like the fuzzy point prototypes,

$$L^i = \frac{\sum_{j=1}^{n} A_i\left(x^j\right)^m x^j}{\sum_{j=1}^{n} A_i\left(x^j\right)^m},$$

and *L* is the central location for the whole data set. As the fuzzy sets A_i form a sub-partition of the given data set, we formulate the problem of determining the optimal direction u as maximizing the ratio

$$\lambda = \frac{\mathbf{u}^T(\mathbf{V}-\mathbf{W})\mathbf{u}}{\mathbf{u}^T\mathbf{V}\mathbf{u}} \quad (0 \le \lambda < 1)$$

Or, in a deferent form, to solve the generalized Eigen value problem

$$(\mathbf{V}\text{-}\mathbf{W})\mathbf{u} = \lambda\mathbf{V}\mathbf{u}$$

Since matrix **V** of the total variance is symmetrical and positive definite, this equation may be rewritten to a matrix equation similar to that obtained in the case of principal component analysis,

$$\mathbf{V}^{-1}(\mathbf{V}\text{-}\mathbf{W})\mathbf{u} = \lambda\mathbf{u}$$

where **λ** and **u** represent the Eigen values (known, as well, as characteristic roots) and Eigen vectors of the matrix **V^{-1}(V-W).** The vector u^1, named the first discriminat factor corresponds to the highest value of **λ**; the higher this value the higher will be the discriminant power of this factor. After obtaining the first discriminant characteristic $\mathbf{c_1} = \mathbf{Xu^1}$, in a similar way can be obtained the discriminant characteristic $\mathbf{c_2} = \mathbf{Xu^2}$, uncorrelated with the first and so on. It appears clearly that eigenvectors corresponding to the matrix $\mathbf{V^{-1}(V\text{-}W)}$ namely $\mathbf{u^1}, \mathbf{u^2}, \dots, \mathbf{u^{k-1}}$, ranked in decreasing order of the positive values $\boldsymbol{\lambda^1}, \dots, \boldsymbol{\lambda^{k-1}}$, are successive solutions of the above matrix equation. The quality of discrimination and the selection of the most discriminant independent variable is given by the value of the largest eigen value, λ.

Finally, the original class means are projected in the new system of coordinates, and the final fuzzy membership degrees are determined from square-distances to the class means, using a relation similar to the Fuzzy C-Means case:

$$A_i\left(x^j\right) = \frac{1}{\sum_{l=1}^{k}\left(\frac{\|x^j - L^i\|}{\|x^j - L^l\|}\right)^{1/(m-1)}}$$

The final fuzzy classification table is computed by counting cardinals of fuzzy sets: instead of counting the number of data items classified in a particular class, we are actually computing an overall fuzzy membership degree. The fuzzy count of all items from the *i*-th original fuzzy set A'_i classified in the *l*-th fuzzy set A_l, denoted as C_{il}, is given by

$$C_{il} = \sum_{j=1}^{n} A'_i\left(x^j\right) \cdot A_l\left(x^j\right)$$

A friendlier version of this fuzzy classfification matrix may be computed by scaling the fuzzy cardinal values and producing values representing the percentages of all items from the *i*-th original fuzzy set classified in the *l*-th fuzzy set:

$$C_{il}^{[\%]} = \frac{\sum_{j=1}^{n} A'_i\left(x^j\right) \cdot A_l\left(x^j\right)}{\sum_{j=1}^{n} A'_i\left(x^j\right)} \times 100$$

A crisp classification matrix is as well determined by first defuzzifying the final fuzzy partition and then using the cardinals of the crisp classes.

After this learning phase, testing follows in various ways, including use of separate testing data, or by cross-validation. The classical discriminate analysis method is known to provide maximum likelihood estimations under certain assumption (normality of the class distributions etc.). As the experiments will illustrate, and as previous research on data analysis methods based on fuzzy sets have also shown, the fuzzy discriminate analysis method is robust with respect to outliers and distribution of data. We underline once again the robustness achieved by using fuzzy membership values. The main advantage of fuzzy sets over crisp sets and of fuzzy logic over binary logic is the availability of nuanced membership degrees. On one side, the classes input provided by the human expert is fuzzified, allowing robust treatment of outliers. On the other side, the output of the method is fuzzy as well, allowing a more detailed view of the relationships between data items and classes. These fuzzy membership degrees are not actually related to uncertainty, because there is nothing uncertain about the classification of a certain data item, but have to be regarded as a measure of 'typicality'.

The fuzzy discriminate analysis method presented here is a multiclass method by design, as no restriction with respect to the number of classes is introduced. This is a parameter to be set by the human experts as they establish the a-priori classes split.

The key idea is that ranges of typing times are assigned to fuzzy sets (e.g., the times in the range of 210–290 milliseconds are part of a set named "very fast"). The sets are called fuzzy because elements can partially belong to a set (e.g., the time 255 is strongly in the "very fast" set while 290 is only weakly a member of the set). In the training phase, the detector determines how strongly each feature belongs to each set, and each feature is matched with the set in which its membership is strongest (e.g. the t-hold-time feature will be matched with the "very fast" set if most t-hold-times are around 255 milliseconds).

In the test phase, each timing feature is checked to see if it belongs to the same set as the training data (e.g., the test vector's t-hold-time is checked for membership in the "very fast" set). The anomaly score is calculated as the average lack of membership across all test vector timing features. Note that we added sets (e.g., "very very fast") to accommodate faster times than seen in the source study (figure 3).

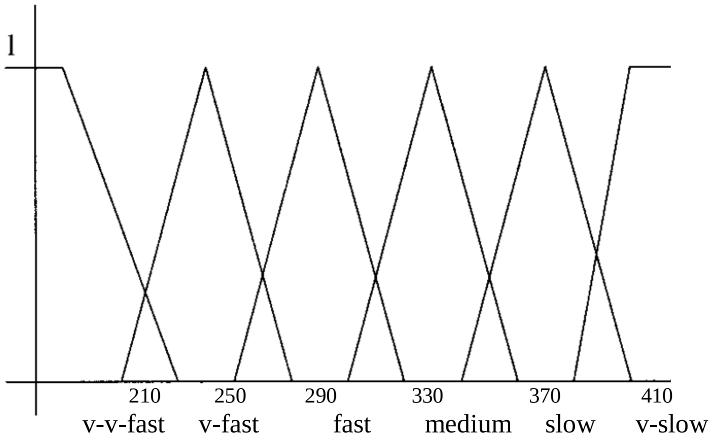

Figure 3: fuzzy set that is proposed

6- RESULT AND CONCLUSION

In this paper, use of typing biometrics and the Fuzzy logic, which acts as an additional security layer for conventional password-based or PIN-based protection systems for computer users, has been investigated. A pressure-sensitive keyboard has been constructed to collected keystroke pressure signals from computer users. In addition to keystroke pressures, keystroke latency signals of the users have been captured. The keystroke pressure and latency signals were presented as the input patterns to the fuzzy logic for it to differentiate between genuine users and impostors.

Voting Fuzzy Logic achieved the best results, with 96.34% Accuracy. Besides, the FMM results compared favorably with those from other classification methods. Although the performance of Fuzzy Logic is good, the standard deviations associated with the results are large. This implies the instability of the Fuzzy Logic performance from one run to another. Thus, further investigation on how to improve the stability of the Fuzzy Logic performance is needed.

REFERENCES

[1] W.E.Eltahir, M.J.E.Salami, A.F.Ismail and W.K.Lai, “Design and Evaluation of a Pressure-Based Typing Biometric Authentication System” Hindawi Publishing Corporation, EURASIP Journal on Information Security Volume 2008.

[2] F.Monrose and A.D.Rubin, “Keystroke dynamics as a biometric for authentication,” Future Gen. Comput. Syst., vol. 16, no. 4, pp. 351–359, 2000.

[3] R.Joyce and G.Gupta, “Identity authentication based on keystroke latencies,” Commun. ACM, vol. 33, no. 2, pp. 168–176, 1990.

[4] D.T.Lin, “Computer-access authentication with neural network based keystroke identity verification,” in Proc. Int. Conf.Neural Networks, vol.1, 1997, pp. 174–178.

[5] Kevin S. Killourhy and Roy A. Maxion. "Comparing Anomaly Detectors for Keystroke Dynamics," in Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN-2009), pages 125-134, Estoril, Lisbon, Portugal, June 29-July 2, 2009. IEEE Computer Society Press, Los Alamitos, California, 2009.