A Multi-Technique Approach for User Identification through Keystroke Dynamics

Sajjad Haider
George Mason University
Fairfax, VA 22030

Ahmed Abbas
Centre for Computer Sc. Studies
Karachi - 75270, Pakistan

Abbas K. Zaidi
Mohammad Ali Jinnah University
Karachi - 75400, Pakistan

Abstract

Legitimate user authentication is an important part of the problems related to the computer and system security. The maintenance of security becomes even more difficult when an invalid user gets the system access information. This paper presents a suite of techniques for password authentication using neural networks, fuzzy logic, statistical methods, and several hybrid combinations of these approaches. The approaches presented in this paper use typing biometrics of a user, in addition to conventional login information, to identify a user.

1 Introduction

Nowadays, one of the most important issues faced by the organizations is to secure their information resources from illegal break-ins and intrusions. These attacks on the organizational information resources can range from physical intrusions to electronic access in data repository through computer networks. Organizations are investing huge sums of money in order to counter such attempts on their vital information resources.

To block the physical access of an invalid user, a number of approaches based on pattern recognition techniques have been developed to verify the user at an entry point of a secured system. Most of these techniques are expensive for small-scale organizations. They require the installation of additional hardware, e.g., face recognition equipment, magnetic card reader, etc.

For accessing a computer almost every operating System uses the approach of assigning a unique username to each of its user, which is also known, to other users of the system. Each user is required to remember a password along with the username. The verification of a user based on username and password is the only mechanism on which access to computer is decided. If an intruder gets the above information of any user then he can become a potential security threat for the information resources. So, there is a strong need to enhance the user verification capability of the existing systems without making such systems unnecessarily expensive.

The fact that the cost is an important consideration, in implementing effective security measures, demands an approach that uses the existing resources. The approach presented in this paper uses the typing biometrics of the user for user validation. Typing biometrics is defined as the analysis of a user's keystrokes patterns [6]. Each person has an almost unique pattern of typing. This pattern can be learned and then can be used for identifying the user. There are approaches suggested in literature [1– 6] for enhancing password authentication by learning the typing pattern of a user. Most of these techniques have several constraints. Some of them require a very long password string. Some of them use a machine dependent approach to learn the typing pattern of a user.

The paper uses fuzzy logic, neural networks, statistical techniques, and the combination of these approaches to learn the typing behavior of a user. The objective is to verify the user based on the learned information and to compare the performance of these approaches.

The paper is organized as follows. Section 2 briefly explains the working of the system and the process of sample collection. Section 3 deals with the terminology related to fuzzy logic, neural networks, and statistics. It also discusses the implementation of these techniques for user verification by learning the typing biometrics of a user. Section 4 presents the results of the analysis. Finally Section 5 concludes the paper and discusses the future direction.

2 Overview of the System

Figure 1 shows the flowchart of the algorithm described. New users are assigned a unique username. They then are required to select a password of their choice. The system limits the password length to 7 characters, which is considered a standard password length. During the learning process users are required to enter the password 15 times. The system is not flexible enough to handle the typographic errors. A user, therefore, is required to enter every password without making errors. The delays between each of the characters of the password string are recorded. Suppose if the password is "qvsldno" then delays

0-7803-6583-6/00/$10.00 © 2000 IEEE

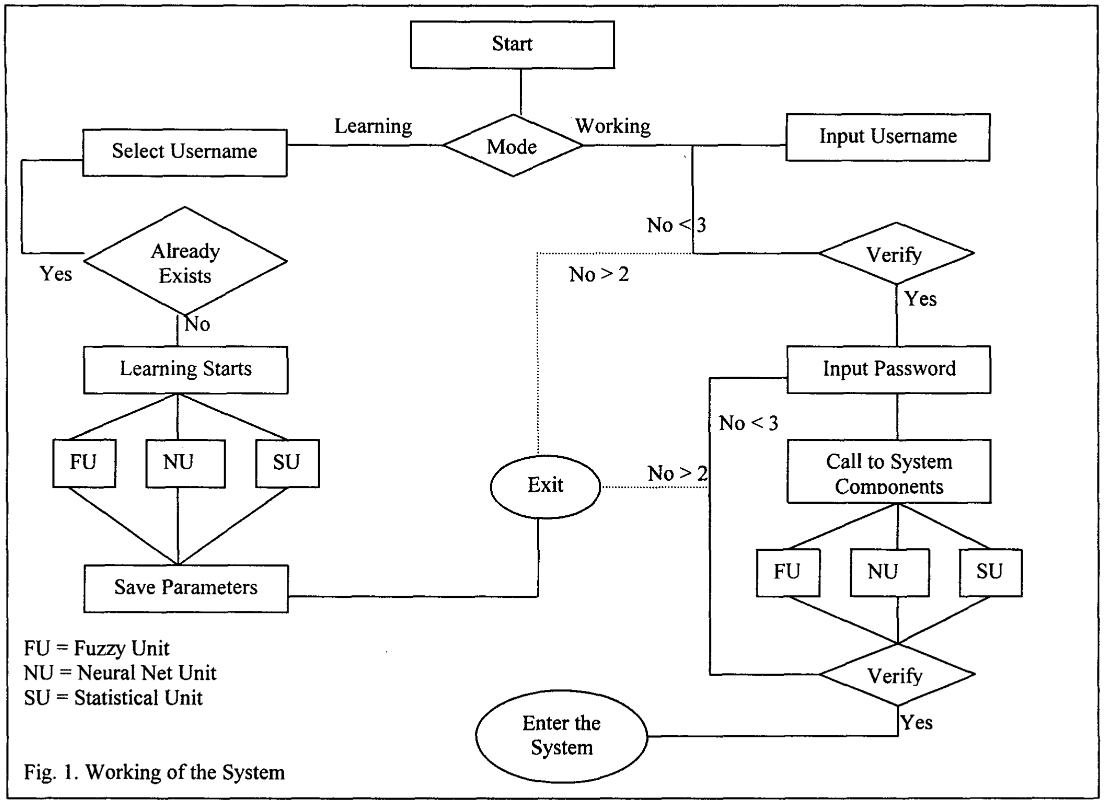

Fig. 1. Working of the System

between (q,v), (v,s), (s,l), (l,d), (d,n), (n,o) are recorded. At the end of the learning process the system has 6 vectors of length 15 representing inter-character delays. These values are then passed to the neural, statistical, and fuzzy unit. They use these values for their parameters setting. The resultant parameters from each unit are stored in the corresponding user profile. The software is developed in C++ and operates in DOS-based environment. The inter-key delays are collected at hundred of a second. The precision can be increased and will result in a better performance.

3 Methods of Approach

This section briefly describes the approaches used in the user verification and their implementation.

3.1 Fuzzy Logic

Several membership functions are used in the system to categorize the typing speed of the user. A sample of these fuzzy classifiers along with their lower and upper bound is shown in Table 1. Figure 2 shows the relationship between these classes.

TABLE 1
Classifiers in Fuzzy Unit

Classifiers	Low Val (10^{-2} sec)	Mid Val. (10^{-2} sec)	High Val. (10^{-2} sec)
Very Fast	21	25.5	29
Fast	26	29	32
Moderate	30	33.5	37
Slow	36	40	44
Very Slow	42	46	50

When the user finishes the password-input process during the learning mode, the collected vectors of password delays are passed to the fuzzy unit. Each element in a particular vector is passed to membership functions. The value is said to belong to a particular category in which it has the largest membership value. The formula for calculating the degree of membership is shown in figure 3. Referring to the input password "qvsldno", the vector v represents the inter-key delay between keys 'v' and 's'

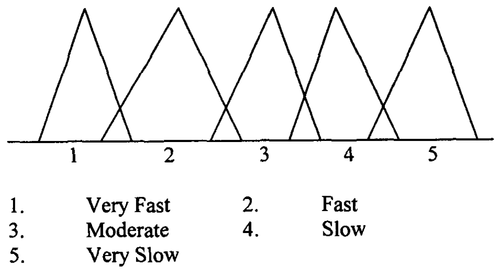

Fig. 2. Fuzzy Categories

obtained during the learning mode.
v = [27, 33, 16, 49, 27, 33, 33, 33, 33, 33, 33, 28, 33, 28, 27] (x 10^{-2} sec)

Each element of vector v is passed to the membership functions. The outcome after the application of these classifiers is shown in Table 2.

```
If (Input < LowerBound OR Input > UpperBound)
Then 0
Else If (Input < MidValue)
    Then (Input – LowerBound) / (MidValue –
LowerBound)
     Else If (Input = MidValue)
          Then 1
      Else (UpperBound – Input) / (UpperBound –
MidValue)
```

Fig. 3. Calculation of Membership Function

Consider the first element of the vector v in Table 2. It has membership in categories Very Fast and Fast with degree of membership 0.57 and 0.33, respectively. Because of the greater degree of membership in category Very Fast, the category Very Fast is selected over there. The process is repeated for each element in the vector. After this process the center of gravity is calculated. Center of gravity, also called the first moment of interia is calculated with the help of (1)

$$I = \frac{\sum \mu_i x}{\sum \mu} \qquad (1)$$

where x is the input value
and μ is the membership value

The value received after the application of center of gravity function is again passed to the membership functions and the category in which this value has the maximum degree of membership is considered to be the representing category for the password-typing pattern between two particular keys.

TABLE 2
Membership Values of Vector x

V (10^{-2} sec)	Very Fast	Fast	Moderate	Small	Very Small
27	0.571	0.333	0	0	0
33	0	0	0.857	0	0
24	0.666	0	0	0	0
49	0	0	0	0	0.25
27	0.571	0.333	0	0	0
33	0	0	0.857	0	0
33	0	0	0.857	0	0
33	0	0	0.857	0	0
33	0	0	0.857	0	0
33	0	0	0.857	0	0
33	0	0	0.857	0	0
28	0.285	0.666	0	0	0
33	0	0	0.857	0	0
28	0.285	0.666	0	0	0
27	0.571	0.333	0	0	0

For vector v, the center of gravity is found to be 31.2. This value is again passed to the fuzzy classifiers and the result is shown in Table 3. Since the center of gravity has the highest degree of membership in category Moderate, the password typing speed between keys 'v' and 's' is considered as Moderate.

TABLE 3
Membership Values of Center of Gravity

CG (10^{-2} sec)	Very Fast	Fast	Moderate	Small	Very Small
31.2	0	0.26	0.34	0	0

The process of finding center of gravity is repeated for all vectors. At the end of this process, following information is obtained and stored in the user profile

1st delay is Slow
2nd delay is Moderate
3rd delay is Fast
4th delay is Fast
5th delay is Fast
6th delay is Fast

This information is used during the working mode to form the fuzzy rule based system, which validates the user. After receiving the inter-key delays of the password during working mode, the fuzzy unit passed these values to the membership functions. The resultant values are matched with the stored information. Table 4 shows the inter-key delays and the resultant fuzzy category when an invalid user entered the same password.

TABLE 4
Fuzzification of Password Delay by an Intruder

Inter-Key Delay (10^{-2} sec)	Category
33	Moderate
27	Very Fast
22	Very Fast
38	Slow
39	Slow
44	Very Slow

The information shown in Table 4 is used by the fuzzy system for the validation of the user. For our example the rule base will match these values against the information stored for the password "qvsldno".

IF (1st Delay is Slow) AND (2nd Delay is Moderate) AND ... AND (6th Delay is Fast) THEN the user is valid.

3.2 Neural Networks

The network used in the system is a 3-layer feed-forward network implementing the backpropagation algorithm. The algorithm uses a supervised mode of learning for memorizing the inter-key delays. There are 6 neurons in the input layer, 4 neurons in the hidden layer and 1 neuron in the output layer. Figure 4 shows the architecture of the neural network.

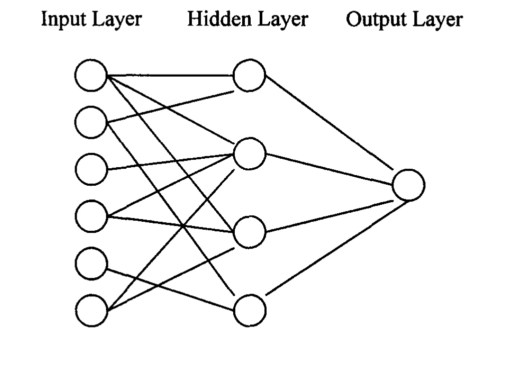

Fig. 4. Neural Network Architecture in the System

At the beginning of the learning process, the weight matrices between input and hidden layer (M1) and between hidden and output layer (M2) are initialized with the values shown. Vectors for hidden neuron biases and output neuron biases are also initialized with same values. The threshold values for input neurons (μ), hidden neurons (λ), and output neurons (γ) are initialized randomly. Sigmoid function used in the network is shown in Eq. 2

$$f(a) = 1 \;/\; (\; 1 + e^{-a}) \qquad \textbf{(2)}$$

$$M1 = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

$$M2 = [\; 0.1, 0.1, 0.1, 0.1]$$

During the learning mode, first vectors of each inter-key delays are averaged and passed as input to input neurons. The value of the output neuron is set to 1. After the initialization of the input and output neurons, learning process is started. After several iterations, when the difference between the calculated output and the desired output is less then the threshold value, the iteration is stopped. The weight of the connections between different layers, i.e., matrices M1 and M2 are stored in the user profile along with the values of the bias vectors and threshold values for all the layers. No negative examples were provided to the system during learning phase.

For the password "qvsldno", input neurons are initialized with the vector y. The vector represents the typing pattern of a valid user.

$$y = [\; 37.5, 31.0, 28.5, 28.8, 29.5, 30.2\;]$$

The output vector is set to 1.

$$z = [1]$$

The network learns the typing pattern of the user after a number of iterations. The final values of M1, M2, hidden bias vector, and output bias vector are stored in the user profile. These values are used to validate the user or to block an intruder. During the working mode the neural net is initialize with these values. Newly entered password delays are passed as an input to input neurons. If the output vector produces the value within a threshold value then the user is considered as a legitimate one.

Considering our example, vector s represents the inter-key delays of the password entered by an invalid user.

$$s = [28, 22, 11, 16, 11, 11]$$

These values are passed to the input neurons. After processing of these values, the difference between the obtained value and desired value at the output vector is greater than the threshold value. Hence the user is declared as invalid.

3.3 Statistics

During the learning mode the statistics unit receives the inter-key delays vectors from the system and calculates the average and standard deviation of each of the set of values. Based on these averages and standard deviations, confidence intervals for each of the delay are formed. Eq. 3 determines the confidence interval for delay 1.

$$x_i \pm z\,\sigma \qquad (3)$$

Where x is the average of the first delay vector,
σ is the standard deviation of the vector,
z is the standard normal distribution value.

At the end of the learning mode, confidence intervals of all the delays are saved in the corresponding user profile. For the password "qsvldno", the confidence interval is shown in Table 5.

TABLE 5
Confidence Interval after the Learning Process

Keys	Lower Limit (10^{-2} sec)	Upper Limit (10^{-2} sec)
(q,v)	20.2	33.1
(v,s)	24.4	37.5
(s,l)	19.6	31.8
(l,d)	20.6	37.1
(d,n)	16.1	31.0
(n,o)	19.7	36.5

When the user enters the password during working mode, the unit loads stored confidence intervals values from the user profile and matches the value of each inter-key delay with the corresponding confidence interval. If a particular inter-key delay lies within the confidence interval then it is assumed as a valid delay. The process is repeated for all inter-key delays entered during the working mode. After matching all the delays the authenticity of the user is decided.

Vector t represents the password-input delays by an intruder.

t = [33, 27, 22, 38, 39, 44]

These values are compared with the confidence interval shown in Table 5. Table 6 shows the result of the comparison. Since only 2 of the values matched with the stored one, the user is considered as an invalid user.

TABLE 6
Matching of Password Delays Entered by an Intruder

Keys	Delays (10^{-2} sec)	L. Limit (10^{-2} sec)	U. Limit (10^{-2} sec)	Match (10^{-2} sec)
(q,v)	33	20.2	33.1	Yes
(v,s)	27	24.4	37.5	Yes
(s,l)	22	19.6	31.8	No
(l,d)	38	20.6	37.1	No
(d,n)	39	16.1	31.0	No
(n,o)	44	19.7	36.5	No

4 Results

During the learning phase, outputs from all three units are recorded. These outputs are used to measure the performance of these units and also the combination of these units. Type I error (probability of being rejected when the user is valid), and Type II error (probability of being accepted when the user is a stranger or intruder) are calculated for each of these combinations. Table 7 shows the result of the experiments.

TABLE 7
Type I and Type II Error

	Type I Error	Type II Error
Fuzzy	0.11	0.19
Neural Nets	0.20	0.22
Statistical	0.02	0.13
Fuzzy, Neural	0.13	0.18
Fuzzy, Statistical	0.02	0.08
Neural, Statistical	0.02	0.14
Fuzzy, Neural, Statistical	0.02	0.06

During the working mode, when the user tries to enter the system, he is given a maximum of 2 chances to enter the password in a correct sequence. It is observed that legitimate users are also rejected by the system in first try but they succeeded when they are given second try. Both attempts are considered as a single unit during our analysis of the result. Initially the system gave only one try to enter the system and Type-I error was very high but after the inclusion of second try, the Type-I error decreased drastically. There was no or very little effect on Type II error of all the combinations. Table 8 shows the Type I value for both cases.

TABLE 8
Comparison of Type I Error

	One Try	Two Try
Fuzzy	0.26	0.11
Neural Nets	0.41	0.20
Statistical	0.03	0.02
Fuzzy, Neural	0.36	0.13
Fuzzy, Statistical	0.13	0.02
Neural, Statistical	0.21	0.02
Fuzzy, Neural, Statistical	0.21	0.02

The system was implemented as a lock control circuit. A circuit was designed to validate the user in our lab. A keyboard was placed outside the room. The user enters the username and password, which is passed by the lock control circuit to the computer running the system through the serial port (RS 232 interface). Figure 5 shows the block diagram of the system.

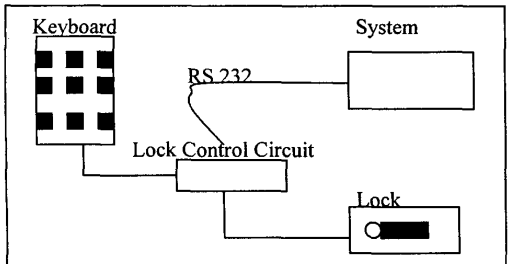

Fig. 5. Block Diagram of the System

5 Conclusion

The paper presents implementations and comparison of various combinations of statistical, neural, and fuzzy techniques for valid user authentication. The system uses inter-key delays of the password for user identification. There are suggestions in the literature, [4 - 5], that a combination of key-hold time with the inter-key delay can improve the performance further. There are some commercially available systems that characterize the password from simple to complex based on the position of characters on the keyboard. An extension of the presented approach that also incorporates these techniques will definitely improve the performance manifold. The increase in precision of the calculated delays may further refine the results.

References

[1] D. Behla, C. Slivinsky, and B. Hussain, "Computer Access Security Systems Using Keystroke Dynamics", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, Dec. 1990

[2] S. A. Behla and M. S. Obaidat, "Computer Users Verification Using the Perceptron Algorithm", *IEEE Trans. Syst., Man, Cybern.*, vol. 23, May/June 1993.

[3] M. S. Obaidat and D. T. Macchairolo, "A Multilayer Neural Network System for Computer Access Security", *IEEE Trans. Syst., Man, Cybern.*, vol. 24, May 1994.

[4] M. S. Obaidat and Balqies Sadoun, "Verification of Computer Users Using Keystroke Dynamics", *IEEE Trans. Syst., Man, Cybern.*, vol. 27, no. 2, April 1997.

[5] J. A. Robinson, Vicky M. Liang, J. A. Michael, and Christine L., "Computer User Verification Using Login String Keystroke Dynamics," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 2, 1998.

[6] Willem G. de Ru and Jan H. P. Eloff, "Enhanced Password Authentication through Fuzzy Logic", *IEEE Expert*, 1997.

[7] H. T. Nauyen and E. A. Walker, "A First Course in Fuzzy Logic", 1999.

[8] S. T. Welstead, "Neural Networks and Fuzzy Logic Applications", 1994.