A Novel Biometric Authentication System using Keystroke Dynamics and Optimized Multilayer Perceptron Neural Network

Priya C V, K. S. Angel Viji

Abstract*: Keystroke dynamics is a peculiar and typical case of biometrics that can be employed to verify the user's individuality. This paper proposes a novel biometric authentication system based on the keystroke dynamics, multilayer perceptron neural network and most valuable player algorithm. The major challenge in machine learning of artificial neural networks is the network training. Due to the nonlinear behavior of the neural network and unknown set of network parameters such as weights and biases, it is more difficult to train the neural network. The most valuable player algorithm is a better alternative to overcome the disadvantages such as local optimum minimum solution and slower convergence speed of the conventional training algorithms. Rapid convergence, proficiency, practicality, and safety are the benefits of the most valuable player algorithm. Hence in this paper, the most valuable player algorithm is proposed to train the multilayer perceptron neural network to overcome the drawbacks in the conventional training process. The proposed biometric authentication system is developed and validated using MATLAB software on different users. The experimental results depict that the proposed method has an authentication accuracy of 99.5917%, which is considered more suitable and efficient for real-time implementation.

Index Terms*: Authentication, biometrics, keystroke dynamics, most valuable player algorithm, neural network.

I. INTRODUCTION

Biometric is an art of recognizing people by a particular biological or behavioral trademark, for example, face, speech, fingerprint, iris, signature, voice and so on [1]–[3]. The singularity of a user biometric can lessen the risk of account theft, and there is no compelling reason for a user to remember or preserve the secret password. Keystroke dynamics is a peculiar and typical case of biometrics that can be applied to check the person's individuality [4]. A special type of biometrics that utilizes the typing rhythm of a character on the keyboard is termed as keystroke dynamics. When the person depresses the keys, distinctive sorts of information can be computed from the person's typing rhythm such as the timing details of the keystroke, finger temperature and pressure [5]. A biometric authentication system utilizing keystroke dynamics is termed as keystroke dynamics based authentication (KDA) system.

The KDA can be executed in two distinct ways: static and dynamic [6], [7].

In the static method, user authentication has been performed in two steps, firstly the password entered by the user is validated against the system database and then the typing rhythm of the correctly entered password is verified [8].

In a dynamic method, called as free text method, the authentication depends solely on the user's typing rhythm irrespective of typed content [9].

The disadvantage of the dynamic method is that the imposter may get authenticated to the secure computer system or app if they try to log on the security system in the different interval with distinct typing rhythms. Whereas in the static method, the user authentication is performed not only by typing rhythm and it also considers the password matches.

Thus, it is very tough to access the security system by the imposter. Therefore, the static way of KDA is a more secure, robust, inexpensive and effective method as compared to the dynamic method.

Hence, the proposed biometric authentication system employs the KDA based on the static model to improve the accuracy and speed of user identification and authentication. Besides, an artificial neural network (ANN) is employed to verify the user via their typing rhythm of the password. The ANN is a smart, intelligent and scientific model motivated by the natural neural system [10]. The multilayer perceptron neural network (MLP-NN) is the most widely recognized and implemented ANN [11]. There are few primary drawbacks in the conventional MLP-NN such as the tendency to converge in local minimum value, very low convergence rate, and highly dependent on initial weight and bias [12]. As alternate options to overcome these drawbacks, train the MLP-NN by meta-heuristic algorithms [13].

Nature is the most significant source for the motivation of meta-heuristic algorithms [14]. A few of metaheuristic methods have earned a prevalence because of good productivity, proficiency and effectiveness, for example, genetic algorithm (GA), simulated annealing (SA), particle swarm optimization (PSO), gravitational search algorithm (GSO) and so on [15]. The sports-based optimization techniques have depicted to be more powerful and robust than the existing methodologies [16]. Henceforth, the primary objective of this paper is to build up an MLP-NN for biometric authentication system using keystroke dynamics and it is quickly trained using a recently developed sports-based optimization technique known as most valuable player algorithm (MVPA) [17]. The MVPA has competitive benefits such as rapid convergence, proficient, practical and dependable [16], [17]. Inspired by these advantages, a novel

Revised Manuscript Received on April 05, 2019.

Priya C V, Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India.

K. S. Angel Viji, Department of Computer Science and Engineering, College of Engineering, Kidangoor, Kottayam, Kerala, India.

hybrid MLP-NN based on MVPA is proposed to train the network and obtain the optimal weight and bias.

After the introduction, the structure and corresponding equations of MLP-NN is discussed in Section II. The MVPA is presented in Section III. Section IV explains the proposed authentication system based on MLP-NN and MVPA. The experimental results and discussions are depicted in Section V. Finally, the conclusions of the proposed biometric authentication system are given in Section VI.

II. MULTILAYER PERCEPTRON NEURAL NETWORK

The structure of the proposed ANN is the MLP-NN [18]. The MLP-NN is an ANN framed of cells mimicking the low-level operations of actual biological neurons. In MLP-NN, the neurons are interlinked in a unidirectional manner. Fig. 1 demonstrates a typical structure of MLP-NN with a single hidden layer. The output of a neuron in the hidden layer is computed using two equations. Initially, the weighted sum of input features that connected to the neuron *j* in the hidden layer is estimated as follows,

$$ws_j = \sum_{i=1}^{l} \omega_{ij} u_i + \beta_j \qquad (1)$$

where u_i is the input feature to the neuron *i* in the input layer; ω_{ij} is the association weight between the neuron *i* and the neuron *j*; *l* is the number of neurons in the input layer, i.e., the number of input features; β_j is the bias corresponds to the neuron *j*.

Next, a transfer function is employed to fire the output of the neuron *j* using the weighted sum of input features calculated in Eq. (1). Further, the output of the neuron *j* in the hidden layer is computed as,

$$f_j(ws) = \frac{1}{1+e^{-ws_j}} \qquad (2)$$

After computing the output value of each neuron in the hidden layer, the inputs to the neuron *k* in the output layer is computed as,

$$y_k = \sum_{j=1}^{m} \psi_{jk} f_j + \delta_k \qquad (3)$$

where ψ_{jk} is the association weight between the neuron *j* in the hidden layer and the neuron *k* in the output layer; m is the number of neurons in the hidden layer; δ_k is the bias corresponds to the neuron *k*. The final output of the neuron *k* in the output layer is estimated as,

$$\hat{Y}_k(y) = \frac{1}{1+e^{-y_k}} \qquad k = 1,2,...,o \qquad (4)$$

where *o* is the number of neurons in the output layer.

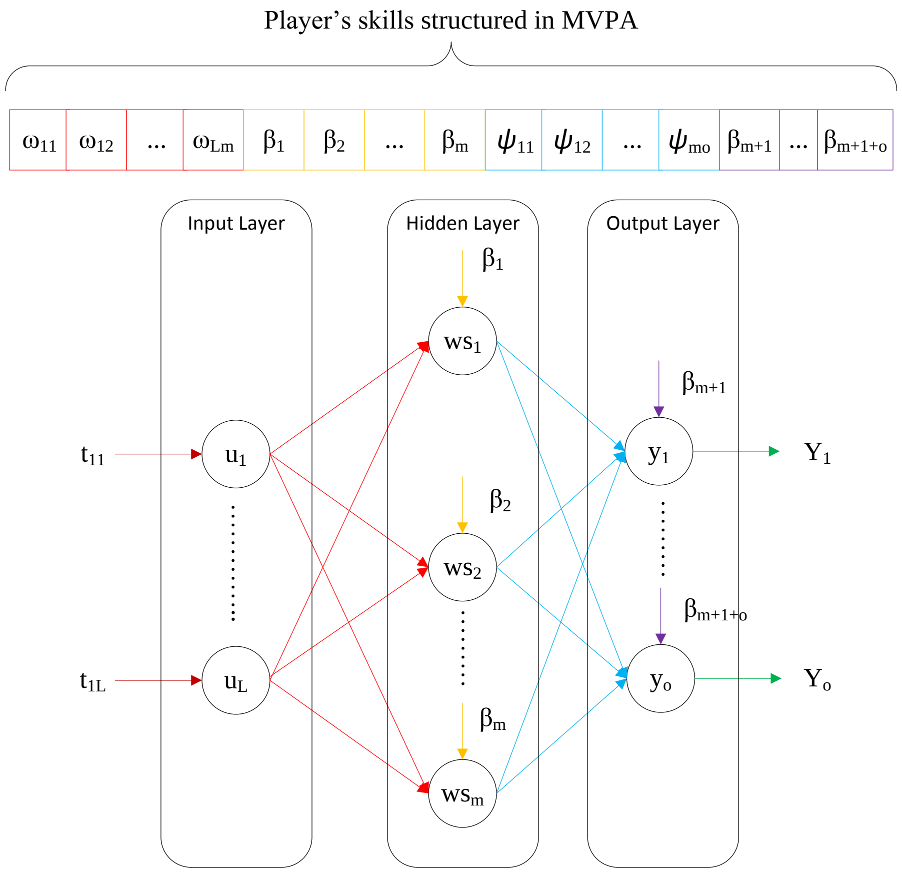

Fig. 1 Structure of MLP-NN with a single hidden layer

III. MOST VALUABLE PLAYER ALGORITHM

Most valuable player algorithm (MVPA) is the recently developed optimization technique based on sports game [17]. The MVPA has some competitive benefits such as rapid convergence, proficient, practical and reliable [17]. The flowchart of the MVPA to train MLP-NN is shown in Fig. 2.

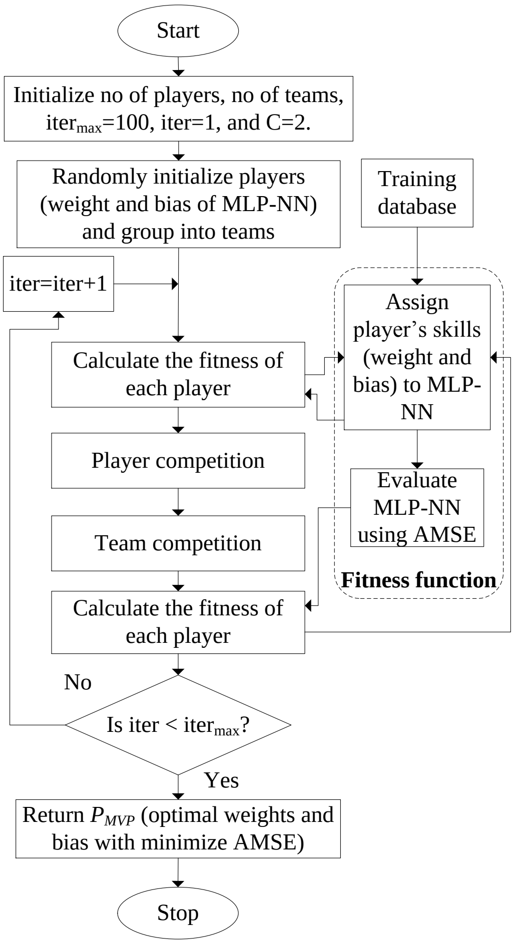
Start
Initialize no of players, no of teams, iter$_{max}$=100, iter=1, and C=2.
Randomly initialize players (weight and bias of MLP-NN) and group into teams
Training database
iter=iter+1
Assign player's skills (weight and bias) to MLP-NN
Calculate the fitness of each player
Evaluate MLP-NN using AMSE
Player competition
Team competition
Fitness function
Calculate the fitness of each player
No
Is iter < iter$_{max}$?
Yes
Return P_{MVP} (optimal weights and bias with minimize AMSE)
Stop

Fig. 2 Flowchart of the MVPA to train MLP-NN

A group of players competes collectively in teams in order to win the leagues' championship, and they compete individually in order to win the most valuable player trophy. The number of players' skills corresponds to the dimension of the problem (MLP weight and bias) and team is composed of a group of players.

First, each player's skills are initialized randomly between the upper and lower bounds of the decision variables. Once the player's skills are organized, all players are arbitrarily deployed to create different teams. Let *ns* be the problem dimension (skills of a player), *np* be the number of players and *nt* be the number of teams participated in the sports league. Therefore, the player P_i can be structured as,

$$P_i = \left[skill_{i,1},\ \ skill_{i,2},\ \ ....,\ \ skill_{i,ns} \right] \qquad (5)$$

The fitness of a player P_i is calculated from the problem's objective function *F(P)* and it is denoted as *fitness*(P_i).

To create teams, the first *nt-1* teams have *np1* players, while the last team τ_{nt} have *np2* players.

$$np1 = floor\left(\frac{np}{nt-1} \right) \qquad (6)$$

where *floor* is a function which rounds the number towards minus infinity.

$$np2 = np - np1 \qquad (7)$$

Therefore, team *j* can be created as,

$$\tau_j = \begin{cases} \left[P_{A_j} \right]_{1\times np1} & if \quad j = 1,2,...,nt-1 \\ \left[P_{B_j} \right]_{1\times np2} & if \qquad j = nt \end{cases} \qquad (8)$$

where,

$$A_j = \left\{ a_1,\, a_2,\, \ldots,\, a_{np1} \right\} \qquad A_j \in randi\left(np \right) \qquad (9)$$

$$B_j = \left\{ b_1,\, b_2,\, \ldots,\, b_{np2} \right\} \qquad B_j \in randi\left(np \right) \qquad (10)$$

where *randi* is a function which returns a pseudorandom real number between *1* and *np*. All sets A_j and B_j are disjoint. Examples to create teams can be seen in the literature [17].

A. Player Competition

Each player aims to be his team's franchise player and the league's most valuable player. That is why, in individual competition step of MVPA, the skills of the players of the selected τ_i are improved as,

$$\tau_i = \tau_i + rand \times (FP_i - \tau_i) + C \times rand \times (P_{MVP} - \tau_i) \qquad (11)$$

where *rand* is a function which returns a uniformly distributed random number between *0* and *1*. FP_i is the franchise player in the team τ_i and P_{MVP} is the most valuable player among all players. *C* is a constant that can be selected to an integer based on the type of optimization problems. In [17], *C* is set as *2* after a few numbers of experimental analyses.

B. Team Competition

A team τ_i plays against another team $\tau_j\ (i \neq j)$. The probability of τ_i beats τ_j is estimated as,

$$pr\left\{ \tau_i \quad beats \quad \tau_j \right\} = 1 - \left[\frac{(fitnessN(\tau_i))^k}{(fitnessN(\tau_i))^k + (fitnessN(\tau_j))^k} \right] \qquad (12)$$

where *fitness*(τ_i) is the fitness or objective value of team τ_i and it is assumed as the fitness of the FP_i in that team. In this study, the exponent *k* is chosen as *1* to estimate the winning probability of a team [17]. *fitnessN*(τ_i) is the normalized fitness value of team τ_i and it is calculated as,

$$fitnessN(\tau_i) = fitness(\tau_i) - min(fitness(\tau \qquad (13)$$

When the team τ_i wins the game, then the skills of players in the team τ_i are enhanced as,

$$\tau_i = \tau_i + rand \times (\tau_i - FP_j) \qquad (14)$$

Else, the players' skills in the team τ_i are improved as,

$$\tau_i = \tau_i + rand \times (FP_j - \tau_i) \qquad (15)$$

In the meanwhile, the newly updated skills of the players are verified for lower and upper boundary limits. If any of the skills violate its lower limit, then the lower limit value is assigned as the new updated skill. Similarly, if the skill violates its upper limit, then the upper limit value is set as the new updated skill.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In the MVPA, there is no tie result. When both teams have the same objective value, then they have the same winning percentage. A uniformly distributed random number is selected to find a winner among the two teams; if the random value is higher than *0.5*, then the first team is declared as the winner, else the second team is the winner [17]. Terminologies used in MVPA are shown in Table I.

Table I Terminologies in MVPA

	Game	MVPA
1	Player	Feasible solution
2	Group of players	Population
3	Skills of players	Design variables (Problem dimension)
4	Franchise player in a team	Optimal solution with best fitness value in a team
5	League's MVP	Optimal solution among the population
6	Fixture	Number of generations or iterations
7	Player fitness	Objective or fitness function

IV. PROPOSED AUTHENTICATION SYSTEM

The proposed KDA system consists of two phases, one is the signup phase (training phase), and the other is the authentication phase (classification phase).

For an illustrative purpose, let us consider a bank's internet banking application. When a new bank customer is interested in registering for personal internet banking, the customer must create their unique username and password on the signup or registration page of the bank website. To generate the training sets for the neural network, the customer needs to type their unique password for a few times. For each customer, a specified number of timing feature sets are extracted from the keystroke dynamics of the password samples. The derived timing features are treated as the training input to the MLP-NN. Two output nodes on MLP-NN are utilized to symbolize legal or illegal user. Further, the MLP-NN is trained by using MVPA to obtain the optimal weight and bias. Therefore, each customer has a specific storage space in the system database where the corresponding username, password, optimal weight, and bias are stored.

A. Timing Features

The typing samples are a simple text entered at a computer keyboard, together with some timing data about the keystrokes. Timing data is typically represented by two primary measures: the time at which a key is depressed and the time at which a key is released. These timing measures are used to compute the timing features, i.e., duration of a keystroke, the latency between two consecutive keystrokes, press-press time, release-release time and press-release time [19].

Key Hold time, H – The time between depressing and releasing a key.

Key Latency time, L – The time between releasing the first key and depressing the following key.

Key Press-Press time, PP – The time between depressing the first key and depressing the following key.

Key Release-Release time, RR – The time between releasing the first key and releasing the following key.

Key Press-Release time, PR – The time between depressing the first key and releasing the following key.

The pictorial explanation of these five keystroke times is exemplified in Fig. 3. In this paper, these five keystroke timings in a valid password are utilized as the training features for the neural network.

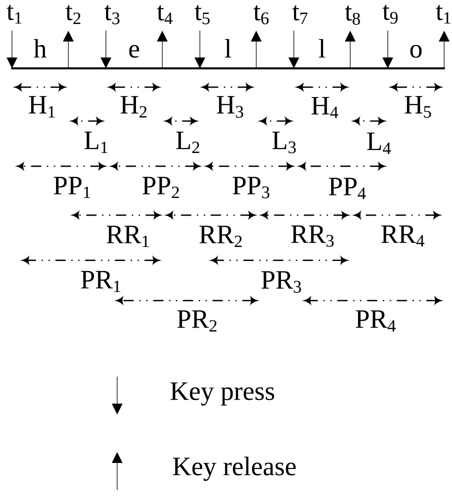

Fig. 3 Keystroke dynamics in typing a five-character password 'hello'.

B. Timing Sets

As discussed earlier, an internet banking application is considered. For each customer, the timing features are collected as the training set for the neural network and it is stored in the training set *T*.

Let *s* be the number of password samples of a user, *n* be the number of characters in a password and then *l* be the number of timing features. Therefore, the total number of features that can be extracted from a password with *n* characters is the sum of the number of *H* features, *L* features, *PP* features, *PR* features and *RR* features. The value of *l* can be computed as,

$$\begin{aligned} l &= n + (n-1) + (n-1) \\ &+ (n-1) + (n-1) = 5n-4 \end{aligned} \qquad (16)$$

For each customer, *T* stores *s* sets of *l* timing features. Thus the matrix size of training set *T* for a user is *s* x *l*. The training set *T* with extracted timing features can be expressed as,

$$T = \left\{ t_{ij} \right\} \qquad i=1,2,\ldots,s\,; \qquad j=1,2,\ldots,l\,; \qquad (17)$$

$$\begin{gathered} T = \left\{ H_{ij},\, L_{ik},\, PP_{ik},\, RR_{ik},\, PR_{ik} \right\} \\ i=1,2,\ldots,s; \quad j=1,2,\ldots,n \quad k=1,2,\ldots,n-1\,; \end{gathered} \qquad (18)$$

where H_{ij} is the hold time of a character; L_{ik} is the latency time between two successive characters; PP_{ik} is the time between depressing the first character and depressing the following character; RR_{ik} is the time releasing the first character and releasing the next character; PR_{ik} is the time between depressing the first character and releasing the following character; $i = 1$ to s, $j = 1$ to n & $k = 1$ to $n-1$.

C. Training Phase

In the presented KDA method, the output of the MLP-NN is binary, i.e., *1* as legal and *0* as an imposter. Thus, the training sets used to train the neural network should contain the timing features of both the legal user and the imposters. As discussed earlier, for each legal user, *s* sets of *ns* training features are available. Therefore, the combined training sets $S = 2s$ are given as the input to the neural network, i.e., *s* sets of input features of a legal user and *s* sets of imposters. Further, the MLP-NN is trained using MVPA.

When developing the structure of the MLP-NN, the selection of the number of hidden layers and the number of neurons in each hidden layer are the primary concerns. There is no standard rule available to choose these numbers. However, the technique presented in [20] considers a single hidden layer MLP-NN with the number of neurons in the hidden layer is computed as,

$$m = 2 \times l + 1 \tag{19}$$

Hence, in this proposed technique, MVPA is implemented to train the MLP-NN with a single hidden layer and m number of hidden neurons. Training an MLP-NN is defined as allotting the optimal weights to each association between two neurons in successive layers and optimal bias to each neuron; with an objective function to minimize the mean square error (MSE) between the actual target Y_k and the estimated output $\overline{Y}_k$. The MSE can be computed as,

$$MSE = \frac{1}{o}\sum_{k=1}^{o}(Y_k - \hat{Y}_k)^2 \tag{20}$$

It is essential for the MLP-NN to fix the unique set of weight and bias values for all the training samples to achieve a minimal error rate. Thus, the average mean square error (AMSE) for all training samples is calculated as,

$$AMSE = \frac{1}{s}\sum_{i=1}^{s} MSE_i \tag{21}$$

Therefore, the fitness function used to train the MLP-NN using MVPA is defined as,

$$Minimize \quad F(P) = AMSE \tag{22}$$

In the application of the MVPA to train the MLP-NN, the vector weight and bias combine to form a feasible solution. In such a way that, each player *P* in the MVPA constituted from four portions; the first one is the connection weights between the input neurons and the hidden neurons, second is the connection weights between the hidden neurons and the output neurons, third is the set of bias on the hidden neurons, and the final portion is the set of bias on the output neurons. The structure of each player's skills is graphically presented in Fig. 1. There are *l* number of input neurons, *m* number of hidden neurons and only one output neuron, i.e., $o = 1$. Therefore, the size of weight-bias vector (problem dimension) *ns* is calculated as,

$$ns = (l \times m) + (m \times o) + m + o \tag{23}$$

MVPA initializes the players' skills (weight and bias) randomly and trains the MLP-NN to determine the optimal global solution iteratively. The optimal solution is achieved by deploying the weight and bias vector on MLP-NN and computing the MSE. The optimal weight and bias vector P_{MVP} identified from the training phase is employed for user authentication.

D. User Authentication using MLP-NN and MVPA

In the authentication phase, when the customer accesses the personal banking application, the username is first searched in the system database. Then the secure system allows the user to move forward to the next page where the password has to be entered. Next, the entered password string is validated in the corresponding user database. If the password matches with the user database, then the proposed authentication framework extracts the timing features from the password string. Further, the extracted features are given as the input to the MLP-NN to classify the user. In the interim, the MLP-NN will retrieve a set of weights and bias P_{MVP} associated with the matched user database and performs network forward computation. The outcome of the classification phase is either legal or illegal.

V. RESULTS AND DISCUSSION

With the assistance of experimental analysis, the behavior of the presented KDA method with respect to the different size of the training set is examined in this section. The various training sample sizes are considered to understand the overall performance of the proposed KDA method. The training database size *S* is ranging from *6* to *100*. The experimental analysis presented in this section gives a significant knowledge of the behavior of the proposed KDA method in practical security applications.

A. Experimental Setup

The proposed authentication system is developed and validated on different persons using MATLAB simulation software in Windows 10 laptop with i7 CPU 1.8GHz, 4GB RAM. The timing features are calculated using five distinct stopwatch timers. Each stop timer has a dedicated interrupt and a timer. The illustration of calculating the hold time and latency time is given as follows,

(i) When a person presses a key, the interrupt in the stopwatch 1 can detect the state of the key press, and it turns "ON" the timer 1. After a fraction of the time, if the person releases the key, the interrupt 1 can sense the state of key release, and it turns "OFF" the timer 1. The time interval between timer 1 "ON" and "OFF" gives rise to the hold time of a character.

(ii) Similarly, the latency time between the two successive characters is calculated using stopwatch 2. When the user releases the first key, the interrupt 2 can detect the state of key release, and it turns "ON" the timer 2. After a while, if the client presses the next key, the interrupt 2 can notify the state of the key press, and it turns "OFF" the timer 2. The time interval between timer 2 "ON" and "OFF" gives rise to the latency time between two successive characters.

Similarly, for all the remaining three timings are computed using three separate stopwatches as like hold and latency time calculations.

B. Performance Indices

Like other biometric-based methodologies, the execution of the proposed KDA method is assessed through different indices [4], [19].

False Acceptance Rate (FAR) – The rate at which an imposter is authenticated as a valid user.

False Rejection Rate (FRR) – The rate at which a valid user is denied to access the secure app or page. A higher FRR shows that the legal user is frequently rejected.

C. Comprehensive Analysis

The proposed KDA method is examined on *100* different legal users to validate the accuracy and robustness of the security system. To calculate FRR, each legal user is allowed to log in to the secure page or app for *100* times at different time intervals. Similarly to calculate FAR, *100* different imposters are permitted to access the secure system by typing the same password.

Table II depicts the experimental results of the proposed security framework for user authentication with different database size *S*. Quite evidently, the overall accuracy of the proposed security framework enhances together with the size of training set increases.

It is possible to notice that FAR appears to decrease when the size of the training samples increases. However, when the user samples are more than *s = 25 (S = 50)*, FAR starts to increase. In addition, the best accuracy of the proposed system is obtained at *S = 50* and FRR is just *0.5056%* which is much nearer to the best FRR of *0.3422%* achieved at *S = 20*. Fig. 4 shows the comparison of FRR and FAR with respect to different database sizes. The inference obtained from Table II and Fig. 4, the suitable training sample size for the presented KDA method is *s = 25 (S = 50)*.

D. Comparative Study

A comparative study has been carried out for the proposed and existing techniques to prove the superiority, proficiency, accuracy and security level of the proposed KDA system. In this study, the proposed KDA method is analyzed for *100* legal users and *100* imposters with each user training set consist of *25* password samples. In [4], an identity verification method based on dynamic keystroke properties had been implemented, and its report indicates that the technique had FRR of *11.1%* and FAR of *12.8%*. A KDA technique had been introduced in [6], and it was verified in both static and dynamic modes with FRR of *0%* and FAR of *15%*. This authentication method utilized the distance measures of keystroke digraphs and examined on *205* different users. The test result depicts that this method has FRR of less than *5%* and FAR is less than *0.005%* [7]. A KDA system using pairwise user coupling and machine learning algorithms has been presented in [21] and the technique has been analyzed on both free text and fixed text inputs. Various case studies were done in online exam based KDA database and the method has achieved an overall accuracy level of *89.7%*. A user-adaptive feature extraction based KDA method has been proposed to improve the user authentication, and the technique utilizes the free text input [19]. Table III depicts the comparative studies of the proposed KDA and the existing methods. From Table III, it is seen that the proposed KDA method has very low FRR of *0.5056%* and FAR of *0.311%* as compared to [4], [6], [7]. These rates are considered as satisfactory for real-time implementation. Generally, the index FRR corresponds to the accuracy level of authenticating a legal user, and the index FAR represents the security level of the system. Therefore, the proposed KDA method with very low FRR and FAR values symbolize that the security system is highly accurate and more secure.

Table II FRR, FAR and overall accuracy of the proposed authentication system

Database size, S	Training time (sec)	AMSE	FRR (%)	FAR (%)	Overall Accuracy (%)
6	3.7576	0.00001297	0.3776	2.3418	98.6403
10	5.2536	0.00001041	0.3773	2.4673	98.5777
20	8.3912	0.00000544	**0.3422**	1.1164	99.2707
30	12.6613	0.00004285	0.8027	0.4852	99.35605
40	16.8608	0.00008776	1.0768	0.7032	99.11
50	20.1941	0.00001371	0.5056	**0.311**	**99.5917**
60	24.043	0.00015593	1.4738	0.4584	99.0339
70	28.1295	0.00013175	1.222	0.3713	99.20335
80	32.0059	0.00012265	1.2232	0.6862	99.0453
90	36.3126	0.00012692	1.2029	0.4891	99.154
100	39.4434	0.00009829	1.1087	0.6303	99.1305

Table III Comparative studies of the proposed and existing methods

Method	FRR (%)	FAR (%)	Accuracy (%)	The speed of authentication (sec)
Ref [4]	11.1	12.8	88.1	55
Ref [6]	0	15	92.5	49
Ref [7]	5	0.005	97.5	33
Ref [21]	-	-	89.7	-
Ref [19]	-	-	99.5	-
Proposed KDA	**0.5056**	**0.311**	**99.5917**	**0.004355**

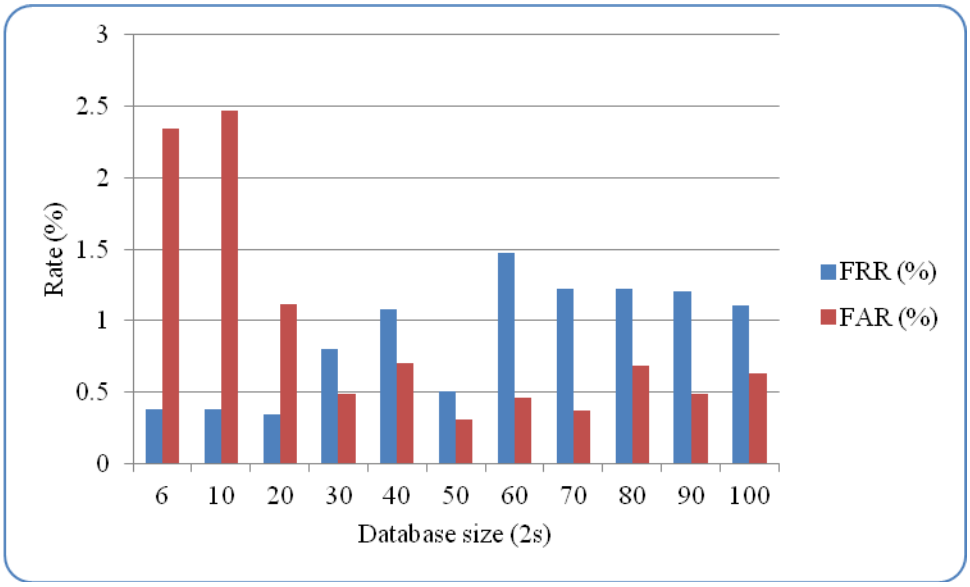

Fig. 4 Comparison of FRR and FAR with different database sizes

The accuracy of the proposed KDA method is about *99.5917%* and which is better than that of the method presented in [19]. Moreover, the proposed KDA method uses the user samples of *25* whereas the technique in [19] utilizes a more extensive database size of *1000*. Hence, the proposed KDA method consumes very less memory to store the training database when compared to other KDA methods. Since the larger training database method consumes more data size to store the lengthy free text content in the system database, and thus the validation process takes too much of time to search and verify over such more massive database, the proposed KDA method can authenticate the user much quicker than the other existing dynamic methods. To verify the authenticating speed of proposed KDA method, user authentication has been performed for *10* times to measure the average time taken to authenticate the user. Both the proposed and existing methods use the same number of sample size for the text content *'hello'*. The average speed of authenticating a legal user in proposed and existing methods is calculated and tabulated in Table III. As discussed before, the proposed KDA method validates a person much faster than other existing techniques.

VI. CONCLUSIONS

In this paper, a novel biometric authentication method based on MLP-NN and MVPA have been developed to authenticate the legal user to access the secure web page or app. The various analyses have been carried out in different environmental conditions, and the test results depict that the proposed KDA method has a more significant security level and very high speed of authentication compared to other existing techniques. The proposed approach has been tested on 100 legal users and 100 unauthorized persons, and it has performed well in authenticating the legal users and impostors. FRR, FAR and authentication accuracy has assessed to show the superiority and robustness of the proposed keystroke dynamics based authentication method. The results show that the proposed KDA method has very low FRR and FAR values when compared to already existing KDA methods. The method has an authentication accuracy of 99.5917%, which is considered as more suitable and efficient for real-time implementation. The proposed feature extraction and neural network training can be employed in both static and dynamic way of authentication.

REFERENCES

1. M J Sudhamani and M K Venkatesha, "Fusion of Iris Texture with Finger vein Geometry for Authentication," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 3, pp. 176–180, 2019.

2. J Ram Prabu, R Pavithra, N Aswini and A F Brindha, “Wireless Smart Biometric Attendance System,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 2s, pp. 156–159, 2018.
3. K. Martin Sagayam, D N Ponraj, J Winston, J C Yaspy D E Jeba, and A Clara, “Authentication of Biometric System using Fingerprint Recognition with Euclidean Distance and Neural Network Classifier,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 4, pp. 766–771, 2019.
4. J. Leggett, G. Williams, M. Usnick, and M. Longnecker, “Dynamic identity verification via keystroke characteristics,” International Journal of Man-Machine Studies, vol. 35, no. 6, pp. 859–870, Dec. 1991.
5. K. Kotani and K. Horii, “Evaluation on a keystroke authentication system by keying force incorporated with temporal characteristics of keystroke dynamics,” Behaviour & Information Technology, vol. 24, no. 4, pp. 289–302, 2005.
6. S. M. Furnell, J. P. Morrissey, P. W. Sanders, and C. T. Stockel, “Applications of keystroke analysis for improved login security and continuous user authentication,” in Information systems security, 1996, pp. 283–294.
7. D. Gunetti and C. Picardi, “Keystroke analysis of free text,” ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 3, pp. 312–347, 2005.
8. R. K. Das, S. Mukhopadhyay, and P. Bhattacharya, “User authentication based on keystroke dynamics,” IETE Journal of Research, vol. 60, no. 3, pp. 229–239, 2014.
9. A. A. E. Ahmed, “Employee surveillance based on free text detection of keystroke dynamics,” Handbook of research on social and organizational liabilities in information security, pp. 47–63, 2009.
10. A. Sento and Y. Kitjaidure, “A Neural Network PID-Like Controller Using a Hybrid of Online Actor-Critic Reinforcement Algorithm with the Square Root Cubature Kalman Filter,” International Journal of Intelligent Engineering and Systems, vol. 11, no. 6, pp. 261–270, 2018.
11. W.-C. Lin, S.-W. Ke, and C.-F. Tsai, “Top 10 data mining techniques in business applications: a brief survey,” Kybernetes, vol. 46, no. 7, pp. 1158–1170, Jun. 2017.
12. I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing connection weights in neural networks using the whale optimization algorithm,” Soft Computing, vol. 22, no. 1, pp. 1–15, 2018.
13. I. Aljarah, A.-Z. Ala’M, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, “Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm,” Cognitive Computation, pp. 1–18, 2018.
14. J. Liu and K. Xie, “Emergency materials transportation model in disasters based on dynamic programming and ant colony optimization,” Kybernetes, vol. 46, no. 4, pp. 656–671, 2017.
15. G. Nagaraju and S. Shankar, “Gravitational Search Algorithm for Power Quality Improvement of WECS with UPQC,” International Journal of Intelligent Engineering and Systems, vol. 12, no. 1, pp. 133–141, 2019.
16. B. Alatas, “Sports inspired computational intelligence algorithms for global optimization,” Artif Intell Rev, pp. 1–49, Oct. 2017.
17. H. Bouchekara, “Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport,” Operational Research, pp. 1–57, 2017.
18. D. Svozil, V. Kvasnicka, and J. Pospichal, “Introduction to multi-layer feed-forward neural networks,” Chemometrics and intelligent laboratory systems, vol. 39, no. 1, pp. 43–62, 1997.
19. J. Kim, H. Kim, and P. Kang, “Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection,” Applied Soft Computing, vol. 62, pp. 1077–1087, 2018.
20. S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Let a biogeography-based optimizer train your Multi-Layer Perceptron,” Information Sciences, vol. 269, pp. 188–209, Jun. 2014.
21. S. Mondal and P. Bours, “Person Identification by Keystroke Dynamics Using Pairwise User Coupling,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 6, pp. 1319–1329, Jun. 2017.

AUTHORS PROFILE

Priya C V is pursuing Ph.D (Part-Time) in Security & Privacy in the discipline of Engineering & Technology, at Noorul Islam Centre for Higher Education (NICHE), Kumaracoil, Tamil Nadu. She is currently working as Asst.Professor/CSE at MES College of Engineering and Technology, Kunnukara, Kerala from 2013. She completed her M.E Computer Science and Engineering at Coimbatore Institute of Engineering and Technology, Coimbatore, Tamil Nadu, (Anna University, Chennai) in 2013 and B.E in Computer Science and Engineering from Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, (Anna University, Coimbatore) in 2011. Her areas of interests are Security & Privacy.

K. S. Angel Viji, Associate Professor and Head, Department of Computer Science and Engineering, College of Engineering, Kidangoor, Kottayam, Kerala has more than 9 years of Teaching and Research Experience and holds doctorate in the area of Medical Image Processing. . She received her B.E. degree in Computer Science & Engineering from Anna University, Chennai in 2005 ; M.E degree in Computer Science & Engineering from Anna University, Chennai in 2007. Her dedicated involvement in R & D in Medical image Processing and Network Security has been recognized through 31 publications amongst which 2 are SCI indexed, 5 are SCOPUS indexed, 11 papers in refereed , indexed Research Journals and the few in indexed conferences such as IEEE, Springer etc. Moreover as a research guide 6 scholars are currently pursuing their research under her direct supervision. She organized Conferences, Research Colloquiums, Faculty Development Programs and Short Term Traiming Programs. In view of sharing knowledge and serve technical community, she became the professional member of IEAE and IEEE. As a part of Continuing education program she has attended many training programs in reputed institutions like IITs and IIMs.