IJERT
ISSN : 2278-0181
International Journal of Engineering Research & Technology
Publish & Find Papers @ www.ijert.org
BROWSE
OPEN ACCESS
Call for Papers

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 6, June - 2014

Keystroke Authentication using Ant Colony Optimization Algorithm

Seham Bamatraf
Information System Department
Faculty of Computers and Information
Cairo University- Egypt

Mohammed Bamatraf
Computer Science Department
Faculty of Science
Hadhramout University- Yemen

***Abstract*—Keystroke dynamic provides a confidence rate for user authentication throughout the work session. In this paper we propose a novel approach to enhance the accuracy rate for user authentication. The approach is based on combing both a sequence alignment and ant colony algorithm. The ACO algorithm is adapted to comprehend the keystroke dynamic variables. The proposed algorithm is evaluated against the traditional classifiers models to measure the accuracy and precision rates of identification.**

Keywords—Keystroke dynamic; security; sequence alignment; Ant Colony Optimization

I. INTRODUCTION

The importance of security in information systems is self-evident. With the development of networks and increase the computers power, enhance the security has become very interesting. However, the traditional user name and password authentication method reflects several shortcomings such as the ability to recognize the password by hackers if it is easy otherwise if the password is more complex the user should to be aware of his password. Although the commonly used biometric authentication technologies, such as finger print recognition and iris recognition have overcome the traditional username and password, many problems were emerged. For example, cost, implementation complexities, and decrease the degree of user's acceptance. However, keystroke dynamic is one of the biometric authentications but provides several advantages that overcome the problems of general biometric technologies. Keystroke dynamics provides higher security with less cost to authenticate users based on their typing behavior on the keyboard or keypad. The behavior is measured using dwell time (time duration to press the key) and flight time (time duration between key up and key down). Keystroke dynamics methods proved a useful tool for user authentication[1]. The method maintains a profile for each user with his features and applies data mining classification techniques to identify users based on their profiles. Different classification techniques have been utilized such as neural network [2], pattern recognition [3], and recently ACO algorithms [4].

Ant Colony Optimization (ACO) algorithms have been derived from the living nature of ants in their colony. In a colony of social insects, such as ants, bees, wasps and termites, each insect usually performs its own tasks independently from the other members of the colony. However, these insects act together without any kind of supervisor or centralized controller to solve the important survival- related problems. ACO [5] [6] is mainly oriented for solving the combinational optimization problems such as travelling salesman person (TSP), vehicle routing, and telecommunication networks. The main purpose is to find good paths through graphs.

Recently, many works have been proposed to exploit ACO in machine learning and bioinformatics areas. ACO can provide an optimization solution in many problems such as learning classification rule. The researchers focused to inspire rules based on the various behaviors of ants. Additionally, ACO is exploited in sequence alignment problems [7]. The sequences are aligned based on ACO algorithm to improve time and quality of alignment process.

In this paper we aim to enhance the efficiency of accuracy of our previous proposed approach [8] for user authentication by using ACO algorithm. However, the previous work achieved high accuracy but suffered from the local-minima trap which is raised from using global sequence alignment. In contrast, ACO was employed in the literature [9] to find global solutions based on its diversity in chosen solutions. For this reasons, in this work we enhance the performance of our previous NM&W keystroke dynamic by exploiting the ACO benefits. In summary, our new proposed approach is divided into several stages starting by converting the keystroke features (dwell time and flight times) of users into sequences. Next we arrange the sequences similar to DNA sequences to find the matches degree of sequences using a customized similarity matrix rather than BLOSUM [10]. However, the matrix is developed from the traditional BLOSUM in parallel with keyboard characters. The final stage of our proposed approach applies the ACO strategy to perform classification and identification users.

The remainder of this paper is organized as follows: Section 2 presents a background about some related techniques and metrics in keystroke and the related works of ACO. Section 3 presents the proposed method. Section 4 presents the experimental and results, and Section 5 concludes the paper.

II. REVIEW AND BACKGROUND

A. Classification Technique

As said previously, keystroke dynamics method maintains a profile for each user with his features and applies data mining classification techniques to identify users based on their profiles. Recent researches create a user profile (template) using a combination of matrices and made analysis statistics to compute the degree of disorder of keystroke latency and duration [11]. Other approach [12] uses a variation of typing sequences. Additionally, a classification technique is also applied to predict the class of users. In essence, classification is considered an important task in data mining approaches and utilized in several commercial and industry applications [13]. In general, the classification task involves with discovering knowledge to predict the class of unknown instances objects in the search space problem. Essentially, the task uses a data set which is divided into training set used to build classification model and test set used to predict classes of unknown instances. The classification model is built by using a classification algorithm which is applied on the training set where their classes are defined previously. For each training instance, the algorithm analysis the relationship between predictor attributes and class. Finally, the algorithm finishes the analysis by discovering a classification model which is applied later to the instances in the test set. The discovered classification model is evaluated according to the predictive accuracy of classes. In the essence, the accuracy of prediction is measured as follows. The predicted class of an instance of test data is validated by comparing by the actual class of the instance. As a result, the predictive accuracy is commonly computed by dividing the number of instances which are correctly classified by the total number of text set instances.

The classification of users was performed based on their templates that were extracted during the acquisition data stage. Several classifications methods were presented include statistical methods to compute the mean and the standard deviation for templates. Early, Joyce and Gupta [14] presented an authentication method based on absolute distance. This method generated error metrics approximately 25% for FAR and 16.36% for FRR. Additionally, Guven and Sogukpinar [15] used vector analysis as a statistics method to authenticate users. Their method achieved about 95% accuracy to authenticate users.

Other proposed methods for classification used neural network. Obaidat and Macchiarolo [16] proved that the hybrid sum-of-products method achieved an efficient identification rate with approximately 97.8% accuracy. Another approach [2] suggested to use weightless neural network for classification. Later, Saevanee and Bhattarakosol [17] used probabilistic neural network and achieved about 99% for identification rate.

Although the accuracy is very high but in neural network it is so difficult to choose the important features in classification. This makes a real problem in continuous keystroke where results are executed online.

The third trend of classification used in keystroke is pattern recognition. This means to identify objects and assign to different categories based on certain algorithms such as the nearest neighbor algorithm. Furthermore, several clustering methods were used like: Bayes classifier, SVM, and graph theory. For example, Giot et al. [18] proposed to use SVM for authenticate users and achieve about 95% identification rate. Recently, Balagani [19] used two classifications: k-nearest and ridge-LR and achieved 87.06% and 93.47% identification rate respectively. Learning methods provide a confidence value to determine a decision and very suitable to identify patterns online.

Finally, the last classification methods contain search heuristic to find an optimum solution. It is preferred in the case of large database. It can also provide multiple solutions. ACO and bioinformatics are examples of such approaches using genetic algorithm. Revett et al. [20] exploited the bioinformatics approach where the authors used multiple global sequence alignment [21] as a classification method to identify and authenticate system users based on their keystroke attributes (used three attributes: di-graphs, dwell times, and tri-graphs). The attributes were discretised into amino (ID/Password). The authentication sequence was then compared with the sequences in the database to get a score using a simple global alignment algorithm. The bioinformatics approach achieves about 0.1% for both FAR and FRR error metrics.

B. Sequence Alignment

This section we briefly give an overview of sequence alignment algorithms which will be used later in our proposed system.

Protein in molecule biology is the main component of RNA which composed DNA. Proteins have complex molecules and play an important role to support life activities: starting from respiration ending with thinking. The schema representation of protein structure involves of a sequence of amino acids connected with each other by a chemical substance. There are about 20 naturally amino acids depending on their specific combination. There are specific genes in all organisims which responsible the main performed functions in the organisiom. Founding the similarity between genes in the organisims has been the main task in the bioinformatics community. Such task refers to the pattern-matching or searching in the computer science view. The main goal is to find the similarity degree between two strings of genes in DNA.

In the essence of bioinformatics community, many works have been proposed to use sequence alignment algorithms to solve the pattern matching problem. A sequence alignment algorithm aims to compute an alignment score to measure the degree of convergence between input sequences. For example, when two sequences A and B are matched by aligned vertically: when a matched occurs at given position, 1 is added; otherwise 1 is subtracted from the whole computing score. Additionally, in the case of global alignment, gaps are introduced when the two sequences matched except some positions. It also serves as a tool to ensure the two strings (sequence) has the

same length. These gaps will significantly reduce the overall alignment score.

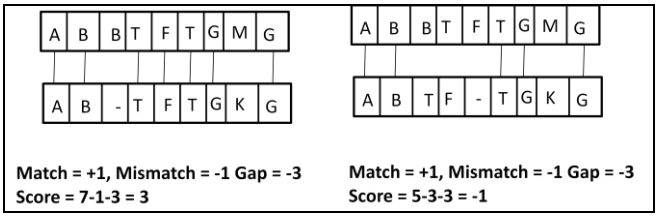

Fig. 1 Sequence Alignment Example

Figure 1 shows an example of two different alignments for two sequences ABBTFGMG and ABTFTGKG. In the alignment on the left-hand, there are 7 matches, 1 mismatch, and 1 gap. On the other hand, the second alignment on the right-hand, there are 5 matches, 3 mismatches and 1 gap. Of course the first alignment is better since it has higher score than other. However, many works have been proposed to determine the best possible alignment. Needleman and Wunch algorithm [21] (NM&W) is an efficient algorithm to determine the best alignment. It was also considered the grandfather of all sequence alignment algorithms to compute the best score. The basic idea starts by the optimal alignment of the smallest possible subsequences (i.e. sequences X and Y have no alignment to each other). Next, an iteration process is started to determine the optimal score by proceeding in both sequences one position at a time. Whenever the iteration process ends, a traceback is started.

```
Input: Two Sequences X and Y
Internal Variables:
       S is a matrix with dimension length of X by length
       of Y
       m: match, mm: mismatch, and g: gap cost
Output: Trace Matrix (TM) for the optimal alignment,
       Computed Optimal score for alignment in the lower
       right-hand corner of Score matrix (S)
1. Initialization:
   1. For i=1 to length(X) S[i][0] = 0
   2. For j=0 to Length(Y) S[0][j]=0
2.  Iteration
   1. For i=1 to Length(X)
      1. For j=1 to Length(Y)
                          S[i-1][j-1] + m(or mm)
            S[i][j]= max{ S[i-1][j] + g          }
                          S[i][j-1] + g

            If(maximum is S[i-1][j-1] + m(or mm))
            Then TM[i][j] = 'Diag'
            else if (maximum is S[i-1][j] + g)
            Then TM[i][j] = 'Up'
            else TM[i][j] = 'Left'
```

Fig. 2 Pseudo Code of Needleman-Wunch Alg.

The traceback process keep tracks the optimal score for each possible subsequence in a matrix. Finally, the optimal score and alignment can be computed easily and efficiently. The traditional pseudo code for NM&W algorithm is shown in Figure 2.

C. An Overview of Ant Colony Optimization (ACO)

Ant colony optimization is a discrete algorithm introduced by Marco Dorigo [5] in the 90's. Next, it is employed to sequential ordering [22] and adaptive for pair-wise sequence alignment [23]. Specifically, ACO is inspired from the behavior of ant colonies. It is scientifically proved that ants are social insects. Their behavior is controlled by biological rules for colony survival rather than individuals. The ants' foraging behavior, especially how ants can find shortest paths between food sources and their nest, has provided the inspiration for ACO. The searching process starts randomly surrounding the ants' nest. While moving, ants leave a chemical pheromone trail on the ground. By investigating this pheromone through smell, ants choose the paths which have been marked by concentrated pheromone. When an ant finds a food source, it analyses the food and takes a sample of it back to the nest. During the return trip, the ant leaves pheromone trail on the ground which indicates the quality and quantity of the food and helps the other ants later to find the source of the food.

In summary, the model reflects the ants' foraging behavior as follows: initially all the ants place in nodes, then each ant moves from source node to food node. Finally, all ants conduct their return trip and reinforce their chosen path as outlined above. The pseudo code of ACO algorithm is shown in Figure 3. The basic idea is to simulate artificial ants to generate new solutions to the given problem. The first step initializes the pheromone. In the second step, ants incrementally build a solution to the optimization problem. Next, the ant evaluates the candidate solution to decide how much deposit of pheromone in the update stage. The update pheromone modifies the pheromone on trails. The modification is either increasing the pheromone if the path is a good solution which will be used in the optimal solution. Otherwise, the pheromone is decreased to allow the exploration of new areas in the search space. Final step is the daemon procedure. This step is daemon action which is a centralized action performed to avoid lies in local optimization. One example of daemon actions is to observe the path that is found by each ant to decide which path is better. Consequently, a set of ants have been selected to perform the deposit additional pheromone on the connection.

While *terminate condition is not satisfied* **Do**
Pheromone initialization
Ant builds a candidate solution in search space
Update pheromone
Perform daemon actions
End While

Fig. 3 Pseudo Code of Metaheuristic ACO

Formally, the designed model [24] is defined as a graph G of network, $G =< V, E >$ where V represents the nodes, and E represents the edges connected the nodes. Two types of nodes distinguished in V include V_s for nest of the ants and V_d for food source node. Edges also are distinguished into two types of links: e_s and e_l connected between nodes V_s and V_d. Here e_s holds the shortest path and e_l holds the longest path between V_s and V_d.

Real ants deposit pheromone on the tracks that walk out. The design of chemical pheromone trail is calculated as follow:

Let τ_i is an artificial pheromone value for two links $e_i, i = 1, 2$, By using the probability:

$$p_{i} = \frac{\tau_i}{\tau 1 + \tau_2}, \quad i=1,2 \tag{1}$$

The ant is able to reach to the food source v_d to choose one of two paths (e_1, e_2).

If $\tau_1 > \tau_2$ then the probability of e_1 is higher,

Else the probability of e_2 is higher.

For returning back to the nest node V_s, each ant will cross the same path with the change of artificial pheromone value τ_i as follow:

$$\tau_i = \tau_i + \frac{\varrho}{l_i} \tag{2}$$

Where ϱ represents a constant value in the model and l_i denotes to the length of the path. The added amount of artificial pheromone is inversely proportional to the length of the path. That is, the shorter of the paths chosen, the higher the amount of added pheromone.

On the other hand, the pheromone trails evaporate over time. This pheromone evaporation is simulated in the artificial model as follows:

$$\tau_i = (1 - \rho) \times \tau_i, i = 1,2 \tag{3}$$

The parameter $\rho \in (0, 1]$ is a parameter that regulates the pheromone evaporation.

III. THE PROPOSED AUTHENTICATION MODEL

In a previous work we addressed the problem of keystroke data representation and classification that improve the accuracy identification of users. In this section we describe theses steps [8] in detail to make the paper self-contained. Next, we discuss the ACO algorithm employment to enhance the performance.

The main data set used for simulation is the set user keystroke dynamics described below. To make sense of this data it is first fuzzified using the membership function given in equation 4. This phase termed as the data representation phase, which is the first phase of the model. Later it is applied to the proposed model described in the subsection below.

A. Data Representation Phase

Features are collected during the login process. These features include the duration of a keystroke, key hold time, and latency. Whenever the data is collected a clean process is started to eliminate any outlier data by taking the median and standard deviation of the data. Next, a fuzzy-logic is used where the data values are represented as a sequence of characters based on the range of speed. For this purpose, we use a membership function and described in Figure 4.

The membership function assigns the range times into fuzzy sets (the times in the range 2-3 msec are part of a set of set *very fast*) and can be represented by character *A.* and so on.

For every user U_i, i=0,..,n-1 where n is the number of users

For every u_i vectors of a user trails set T_j is constructed and mapped to T'_j, j=1,.., m+1 where m is the number of characters

$T'_{ji} \in \{A\ B\ C\ D\}$ where A, B, .. are fuzzy values

The mapping process is performed as following:

$\forall T'j$ the values are as mapped using $f(x)$:

$$f(x) = \begin{cases} A & \text{if} \quad 2 \le x < 3 \\ B & \text{if} \quad 3 \le x < 4 \\ C & \text{if} \quad 4 \le x < 5 \\ D & \text{if} \quad 5 \le x < 6 \\ \ldots \end{cases} \tag{4}$$

Fig. 4 Pseudo Code of Membership Function

In the case of large ranges such as [2-8] msec, the range is assigned into 6 fuzzy sets as can be evidently computed from Equation (4). Each character is measured based on the gap that exists between a pair of characters. For example, the "A" has a minor difference from letter "B" compared with far letter such as "D". Such consideration make an important role when computing the penalty and the score using NM&W algorithm as will be described in next section.

At the end of this phase, we get a set of resulted sequences which are then used as unique patterns for each user (reference signature).

B. Computing Sequence Similarity

In this phase, NM&W algorithm is employed to compute the similarity degree between users. Before using NM&W algorithm, we create a customized matrix to compute similarities rather the traditional BLOSUM matrix. The new matrix consists of the arrangement alphabetic letters for rows and columns. Each intersection is assigned by a value represents the similarity degree between the intersected pair letters. Several measures can be considered such as the alphabet arrangement, distance between letters in the keyboard, and so on. Here we use the alphabet arrangement for example the similarity degree between letters A and A takes the largest similarity degree, and the similarity degree between letters A and B will be less than (A, A), however the degree of similarity for letters that are far then their similarity will be less.

Consider the two strings keystroke fuzzy sequences to be globally aligned are:

$$q_i \ldots\ldots sequence1, i = 0..n$$
$$q_j \ldots\ldots sequence2, j = 0..m$$

Each sequence consists of a set of alphabet characters. Three steps are performed to compute the score using classical NM&W algorithm with slight modification includes initialization, scoring, and traceback. The initialization step involves creating an *n* by *m* scoring matrix and initialized by multiplying each cell by the value of gap penalty. The scoring step compute the cells based on three

cells up, left, or diagonally. Finally, traceback step is conducted to find the optimal sequence alignment.

C. Proposed System for Keystroke using ACO

The proposed model is innovative rather inventive. The model employs the ACO algorithm as the core element in the proposed model. Unfortunately, ACO algorithm in its original construct can't handle all types of data. The proposed model reconstructed the ACO algorithm to handle the fuzzified keystroke sequences from one side; it also introduced a customized selection and pheromone update based on the NM&W alignment algorithm as shown in Figure 5.

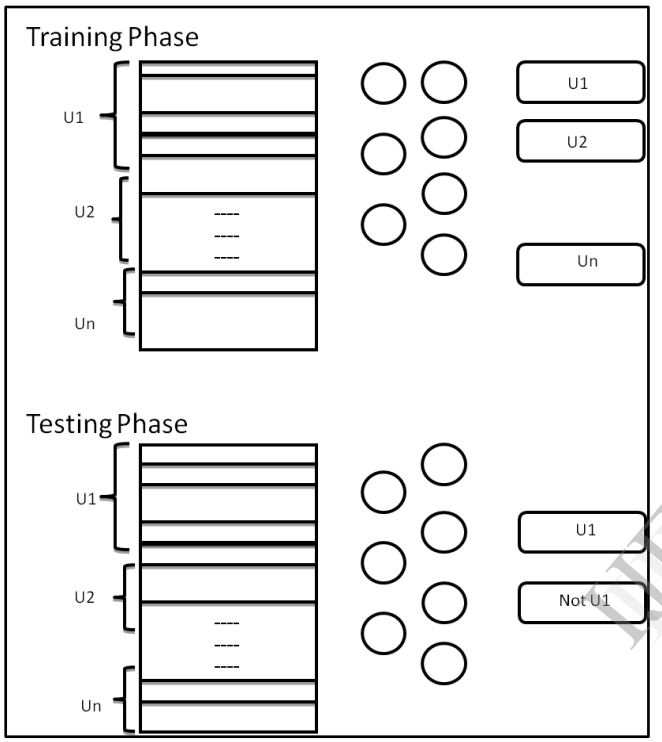

Fig. 5 Keystroke-ACO based

As shown in Figure 5 the core structure is the ACO. Nodes represent user trials, and the routes are decided on the generated score by every ant, the pheromone value is updated accordingly. Firstly, nodes are initially set to random unlabeled values (sequences/vectors), from the training set. Then a set of ants represents user trials set are used to train the system and select best routes for the user.

Ants usually cooperate of communicate by pheromones. Traditional ACO algorithm uses the probability measure to check the level similarity to select a specific path formula equation (1).

Here, the decision is made based on the similarity score between the ants and nodes using the set of steps of sequence alignment of NM&W algorithm which is described in previous section. The pheromone is also updated similarly using equation 2 in Section II.

D. Training and Rule Extraction

For the graph $G =< V, E >$ where V is consists of two sets of nodes namely V_a and V_f. V_a represents the set of ants, r is the number of exemplars or trials for the target user. V_f is the destination or food node set, where r is a selected number estimated based on the number of target rules that could represent the class, the minimum set of nodes with high entropy and can represent the class are selected to be the classification rule set for the class.

E is the set of edges connects V_a'set say e_1, and e_2 is an edge followed to reach a target V_f. The set of edges E receives the pheromone from the ants while ant movement. Thus, the pheromone values are modeled as follows.

For a pheromone value τ_i for every pair of V_a's this value indicates the strength of the target pheromone on the target path.

```
For every  Va in the network:
      As a start point every ant is set to an
      exemplar  xi,  an  ant  chooses  with
      probability as Eq(1).
      The pheromone value is set based on the
      similarity score between  nodes  as
      explained on the steps below:

      The  amount  pheromone  that  is  added
      depends on the average score value of the
      chosen  path:  the  path  with  high
      similarity,  high  score  indicates  more
      pheromone. The used pheromone update used
      is  identical  to  Eq(2)  in  the  classical
      ACO algorithm described in section II.

      Otherwise,  the  fewer  score
      generated  or  unvisited  paths,  are
      subject to pheromone evaporation as given
      in  the  equation  (3)  as  discussed  in
      classical ACO section.
End For
```

Fig. 6 Pseudo Code of Keystroke authentication using ACO System

The pseudo code for adaptation of ACO algorithm in keystroke dynamic is shown in Figure 6. Here no return trip is conducted from the ants, which will definitely safe time.

Finally, the optimal set of solutions represents the set of rules can be used to identify the user. This process is iteratively repeated for every user, coming up with set of rules to be tested and evaluated as discussed in the experimental results section.

IV. The Experiments and Results

In the section we investigate the impact of using ACO to enhance user authentication in our purposed model. For this purpose we will conduct a set of experiments to measure the accuracy and precision of authentication against the stat-of-the-art classification methods.

A. Settings

The experiments used the benchmark data set which are presented by Killourhy and Maxion [25]. The data set was composed of timing information that were collected from 51 users (typists) each of which types a specific

password for 400 times over 8 sessions. Finally, a password time-table was generated from the users' keystrokes and timestamps. The table contained 34 columns and 20400 rows (50 repetitions for 51 users over 8 sessions).

The database is divided into 50 partitions for each user and our proposed model is run using 5 partition as test set and 45 as training set. The data are represented as a sequence of characters to be ready applied in our proposed model.

For the purpose of comparison with our proposed model, various classifiers in the state-of-the-art which have different categories are applied to the same database using Weka tool.

The accuracy and precision rates are computed through experiments to evaluate the effective of different classification methods. Such accuracy and precision are computed according to the following equations:

$$accuracy = \frac{TPos + TNeg}{TPos + TNeg + FPos + FNeg} \times 100 \quad (5)$$

$$precision = \frac{TPos}{TPos + FPos} \times 100 \quad (6)$$

Whereas TPos represents the number of users who are truly accepted, TNeg reflects the number of illegal users who are not accepted. Meanwhile, FPos is the number of illegal users who are accepted. FNeg indicates the number of legal users who are not accepted.

All experiments are conducted on a PC running windows 7 with RAM 3 GB and CPU 2.2GHz. ACO algorithm is downloaded from [26] and we modified it to adapt our proposed keystroke dynamics method using visual studio c#.

B. Results

The accuracy and precision of our proposed system with ACO-based and our other traditional classification methods are listed in Table I. From Table 1, it can be seen that the accuracy rate of our both proposed models with and without ACO are more effective than other classifications methods. The precision rate also has the same noticeable effect. This is due to several reasons. First, the traditional classification methods did not work well with the nominal data which using in our experiments. However, the performance may different when using the real data. Second reason also concern with the traditional classification methods which are based on the whole attributes of the users. This leads to reduce the chance to find the similarity between users.

Table I Classifiers Results using Weka Tool

Method	Precision	Accuracy
BayesNet	50.5	50.5
NaiveBayes	50.8	50.7
Kstar	46.8	47.2
Id3	35.15	35.2
J48	39.7	40.12
Nnge	40.7	40.99
Decision Table	31.41	27.5
Conjunctive Rule	2.1	7.51
Enhanced NM-W Alg.	90.3	80
Enhanced ACO Alg.	91.7	82.2

Other reasons concern with our proposed systems where a heuristic search is used to find more optimum solutions and avoid the local-minima trap that is raised by using NM&W algorithm in our previous method. This explains the superiority of our proposed system with ACO above the NM&W alone.

I. CONCLUSION

In this paper, we explain using of ACO in user authentication based on keystroke dynamic. The proposed system consists two phases: training phase and testing phase. During the training phase, we construct a classifier model by customizing a graph for each user and generating sequence nodes to compute score using NM&W algorithm. In the testing phase, users are authenticated based on the classification rules constructed in the training phase. The proposed system improved performance by 41 % and 32% for precision and accuracy compared to the traditional classification models and about 1% and 2% compared to our previous classification method without using ACO.

References

[1] S. K. Kevin and A. M. Roy, "Free vs. transcribed text for keystroke-dynamics evaluations," presented at the Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results, Arlington, Virginia.

[2] S. Yong, W. Lai, and G. Goghill, "Weightless Neural Networks for Typing Biometrics Authentication," in *Knowledge-Based Intelligent Information and Engineering Systems*. vol. 3214, M. Negoita, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 284-293.

[3] K.-s. Sung and S. Cho, "GA SVM wrapper ensemble for keystroke dynamics authentication," presented at the Proceedings of the 2006 international conference on Advances in Biometrics, Hong Kong, China, 2006.

[4] D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification With Ant Colony Optimization," *Trans. Evol. Comp*, vol. 11, pp. 651-665, 2007.

[5] M. Dorigo, G. Di Caro, and L. Gambardella, "Ant Algorithms for Discrete Optimization," ed, 1998.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents," *Trans. Sys. Man Cyber. Part B*, vol. 26, pp. 29-41, 1996.

[7] Y. Chen, P. Yi, C. Juan, L. Wei, and C. Ling, "Partitioned Optimization Algorithms for Multiple Sequence Alignment," in *20th International Conference on Advanced Information Networking and Applications, 2006. AINA 2006.*, 2006, p. 5 pp.

[8] S. Bamatraf, M. Bamatraf, and O. Hegazy, "Keystroke Authentication on Enhanced Needleman Alignment Algorithm," *Intelligent Information Management*, vol. 6, p. In Press, 2014.

[9] M. D. Toksari, "Ant Colony Optimization for Finding the Global Minimum," *Applied Mathematics and Computation*, vol. 176, pp. 308–316, 2006.

[10] S. Henikoff and J. G. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," in *Proceeding of the National Academy of Sciences of the United States of America* vol. 89, ed, 1992, pp. 10915–10919.

[11] D. Rudrapal, S. Das, and S. Debbarma, "Improvisation of Biometrics Authentication and Identification through Keystrokes Pattern Analysis," in *Distributed Computing and Internet Technology*. vol. 8337, R. Natarajan, Ed., ed: Springer International Publishing, 2014, pp. 287-292.

[12] Z. Syed, S. Banerjee, and B. Cukic, "Leveraging Variations in Event Sequences in Keystroke-Dynamics Authentication Systems," in *High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on*, 2014, pp. 9-16.

[13] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*: Elsevier Science, 2005.

[14] R. Joyce and G. Gupta, "Identity Authentication based on Keystroke Latencies," *Commun. ACM*, vol. 33, pp. 168-176, 1990.

[15] A. Guven and I. Sogukpinar., "Understanding users' keystroke patterns for computer access security," *Computers & Security*, vol. 8, pp. 695 – 706, 2003.

[16] M. S. Obaidat and D. T. Macchiarolo, "An online neural network system for computer access security," *Industrial Electronics, IEEE Transactions on*, vol. 40, pp. 235-242, 1993.

[17] H. Saevanee and P. Bhattarakosol, "Authenticating User Using Keystroke Dynamics and Finger Pressure," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, 2009, pp. 1-2.

[18] R. Giot, M. El-Abed, and C. Rosenberger, "GREYC keystroke: A benchmark for keystroke dynamics biometric systems," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on*, 2009, pp. 1-6.

[19] K. S. Balagani, V. V. Phoha, A. Ray, and S. Phoha, "On the discriminability of keystroke feature vectors used in fixed text keystroke authentication," *Pattern Recogn. Lett.*, vol. 32, pp. 1070-1080, 2011.

[20] K. Revett, "A Bioinformatics Based Approach to Behavioural Biometrics," in *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, 2007, pp. 665-670.

[21] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology* vol. 48, pp. 443--453, 1970.

[22] J. Thompson, D. Higgins, and T. Gibson, "Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position Specific Gap Penalties and Weight Matrix Choice," *Nucleic Acids Research*, vol. 22, pp. 4673-4680, 1994.

[23] L. Dong and H. Hong—wei, "An Adaptive Ant Colony Optimization Algorithm and Its Application to Sequence Alignment," *Computer Simulation*, vol. 22, pp. 100-106, 2005.

[24] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*: Oxford University Press, Inc., 1999.

[25] K. Killourhy and R. Maxion. (2009, 29/06/2014). *Keystroke Dynamics - Benchmark Data Set*. Available: http://www.cs.cmu.edu/~keystroke/

[26] J. McCaffrey. (2012, 6/26/2014). *Test Run - Ant Colony Optimization*. Available: http://msdn.microsoft.com/en-us/magazine/hh781027.aspx