PaSSIL: A New Keystroke Dynamics System for Password Strengthening Based on Inductive Learning

SALEH M. ABU-SOUD
Department of Software Engineering
Princess Sumaya University for Technology
P.O. Box (1438) Amman 11941
JORDAN
abu-soud@psut.edu.jo

Abstract: - In this paper, anew inductive learning system, called PaSSIL, has been developed aims at strengthening the security of systems through keystroke dynamics against both online and offline attackers. The proposed system involves collecting data that represents the biometric patterns of users and converting this biometric data into features that can be manipulated by our inductive learning algorithm called ILA that produces inductive rules that are, in turn, used to classify the legitimate user (called the owner) from adversaries. This system is non-static and since features are a function of the user and the environment, it takes into consideration the gradual changes that happen in the way a user enters his password. Experiments have been conducted on data collected from hundreds of users and compared with our previous work and with some well-known systems. The results obtained are comparable with other systems if not better.

Key-Words: - Keystroke Dynamics, Inductive Learning, ILA, Biometrics, Security

1 Introduction

More than ever before, people depend heavily on computers and internet. This increasing dependency brought to our attention the necessity of safeguarding the huge amount of information inside our systems. In addition, accessing systems globally makes the traditional ways such as passwords and PINs are no longer adequate for protecting our information and computer resources from illegal access, and makes it necessary to find out advanced ways for protection, such as keystroke dynamics.

Keystroke dynamics may be considered as a natural choice for computer security [3]. It is based on characterizing users according to their way of typing on a keyboard. This ideally means that, intruders cannot access the system even if they know the password. This idea stems from the fact that each person has a unique habitual rhythm patterns in the way he types.

Keystroke dynamics are in general preferred because they provide an extra level of security over traditional methods such as passwords and ID cards. These techniques have more desirable properties over some of the identifying biometric features being used as identification based systems such as retina, finger prints, and voice prints, because these techniques are too expensive to deploy, need extra hardware and sometimes easy to fool, while techniques based on keystroke dynamics are user-friendly, non-intrusive, and cost-effective mechanisms. They are easy to implement since they need only a computer program and a keyboard.

Classification among users is done through a group of features that are used to characterize individual keystroke dynamics. The feature that have more distinguishing information for user authentication in the best. These features are usually extracted using the timing information of the key down/hold/up events. The most commonly used features may include: duration which is the hold time of a key, digraphs and time interval between two successive keys [1, 2, 3], trigraphs [3] which are the time latencies between every three successive characters, and speed which is the overall time for entering the whole password.

Keystroke dynamics classification utilizes many classification techniques, that may range from statistical methods [4, 5, 6, 7], to distance based classification which uses many distance techniques such as Mahalanobis distance [8, 9], Manhattan distance [10, 11], and Euclidean distance, and machine learning approaches, which include K-means methods [12], Fuzzy logic [13], Bayesian classifiers [3], K-Nearest Neighbor classifiers [14, 15], Boost learning [16], and Support vector machines [17, 18]. Leggett and his colleagues [20] suggested a keystroke system as a means of dynamic identity verification, in which, a verifier

based on dynamic keystroke characteristics allows continuous identity verification in real-time throughout the work session. Another approach uses neural networks [9, 19, 13, 30, 21, 23, and 24] which has proved itself to be a general learning paradigm for a variety of applications. For instance, Bleha and his colleagues [22] has applied neural networks using standard backpropagation to keystroke dynamics, generating error rates on the order 2-4%. K. Revett et al. also deployed probabilistic neural network as the authentication technique through the PNN algorithm [34]. They also applied a modified version of their PNN algorithm that used separate smoothing factors for each class [35].

A non-static biometric technique; called DLO [6] has been developed. This system is a statistical model aims at identifying users based on analyzing the habitual rhythm patterns in the way they type. This method is based on constructing a hybrid model by exploring the best combination of three keystroke metrics; press duration, latencies between keystrokes and key even order, jointly in one model. The experiments conducted showed that the results of this approach are more accurate than if these measurements are adopted separately. So one of the main aims of this model is to study and examine the best combination of these measurements in one model.

In a newer model which is based on DLO; called DLOS [7], a new metric is added which is the speed of typing the overall password, to the previous ones. In addition, the way of computing some measurements in DLO were modified. The experiments showed that DLOS recorded significant improvements over DLO.

DLO and DLOS are two statistical models based on a sequence of statistical calculations on the desired password. These calculations are applied on data collected from the owner of the system and some adversaries; called in these models user X and adversary user X consequently.

In DLO and DLOS, data is collected once before starting the calculations, so if the owner changes his way of entering the password then he will be refused from accessing his system. Another problem in such models is that they are built on data collected from a limited number of persons. This produces weaker models with significant limitations and capabilities and cannot be generalized.

Actually, this is the main aim of this paper in which a new keystroke dynamics system for password strengthening based on inductive learning; called PaSSIL is proposed and examined. PaSSIL utilizes a powerful inductive learning algorithm called ILA [27]. PaSSIL has proved that it is a powerful and general system with results comparable to other systems if not better. This is achieved mainly because the fact that PaSSIL aims at focusing on the following points that makes it distinguished from other systems in the domain:

1. Most systems are based on several users (10 to 15) to test their capability of distinguishing between the owner of the system and intruders, while PaSSIL utilizes machine learning techniques on data collected from a huge number of users to build the rule base that is used to recognize the owner from intruders, that contributes in enhancing the accuracy of the system significantly.
2. As it is known, a user may change his pattern of entering the password gradually. This is caused by changing the moods, atmospheres, situations, emotional state, stress, and drowsiness, etc. [25, 26]. Other systems do not take into account this into account, while PaSSIL does. So, for instance, if the owner has been injured and cannot write in the same way, the system will correct its behavior according to the new behavior of the owner and; after several attempts, it will recognize the owner correctly again.

ILA and the proposed system are discussed in the consecutive sections.

2 The Inductive Learning Algorithm (ILA)[1]

ILA, which was originally used to connect Decision Support Systems with Expert Systems [28] and fully discussed as a powerful standalone inductive learning algorithm in [27], is an inductive algorithm for generating a set of classification rules for a collection of training examples. The algorithm works in an iterative fashion. In each iteration, it searches for a rule that covers a large number of training examples of a single class. Having found a rule, ILA removes those examples it covers from the training set by marking them and appends a rule at the end of its rule set. In other words ILA works on a rules-per-class basis. For each class, rules are induced to separate examples in that class from examples in all the remaining classes. This produces an ordered list of rules rather than a decision tree. ILA has many advantages; firstly, the rules are in a suitable form for data exploration; namely a

[1]An implementation of ILA can be found in: https://www.researchgate.net/profile/Saleh_Abu-Soud

description of each class in the simplest way that enables it to be distinguished from the other classes, secondly, the rule set is ordered in a more modular fashion which enables to focus on a single rule at a time. ILA applies stepwise forward technique to select an equivalence block of feature(s) included in that class, which provides the basis for generating certain rules.

ILA is designed for handling discrete and symbolic attribute values in an attempt to overcome the attribute selection problem. Continuous-valued attributes can be discretized during decision tree or rule generation by partitioning their ranges using cut points. Many variations for ILA have been developed such as DCL [29], PILA [30], ILA2 [31], and DRILA [32].

3 The Proposed System: PaSSIL

The proposed system is composed of five stages as follows: Determination of parameters, Collecting Data, Discretization, Applying ILA to Produce Rules, and finally Classification. These stages are discussed in details in the following paragraphs:

a. Determination of features:

Since keystroke biometrics indicates that each user types in a uniquely manner with unique characteristics, these characteristics are captured and fed into ILA to produce rules that distinguish the owner from intruders, so that if an intruder tried to hack the rightful owner's account, he/she should be denied access. According to what has been stated previously, three features are being implemented and studied in our proposed model by which a user can be authenticated. These are:

i) Key press duration:

The key press duration is a technique used to identify users by calculating the time interval between the key press and the key release for each letter in the password. It is the duration of each keystroke, i.e. how long is the key held down.

ii) Latencies between successive keystrokes:

Latency is defined as the time interval between a consecutive pair of keystrokes. In other words, it calculates the time interval by which a user releases the first letter and presses the second letter in a certain pair of keystrokes.

iii) Password overall typing speed:

Overall password typing speed measures the time from the key press of the first password character to the key release of the last character. For simplicity, this metric henceforth will be called Speed. Actually, these three parameters have been considered here because it has been noted that they are convergent for the same user while they are divergent among different users and thus can be used as effective metrics to classify users [6] [7].

So, for a password with n characters length, we will have 2n+1 classes: n classes for key press duration for each character, n-1 classes for latency between adjacent pairs of keystrokes, 1 class for speed, and 1 class as a decision class. For example for the password "high" we need 9 classes as follows: h duration, i duration, g duration, h duration, hi latency, ig latency, gh latency, speed, and decision.

b. Collecting data:

Data are collected, through special software built for this purpose, from the owner and a lot of volunteers as adversaries to the system. This data represents the biometric rhythm of the persons who entered the password, and converting this biometric data into a form that can be manipulated by ILA to produce rules that defines the users' biometric traits, against which he can then be authenticated in future. The proposed model aims at developing a system that has low rate of rejecting the owner of the account to enter his system and at the same time to minimize or hopefully to totally prevent adversaries from entering the system even though they know the password.

The collected data was collected using the same keyboard with different times with the assumption of no mistakes are allowed, and is measured by Milliseconds (ms) and hence it is continuous data that needs to be converted to discrete values in order to be applied by ILA. This process is fully described in the following point.

c. Discretization:

Discretization is the process of converting continuous data into discrete ones without losing the value and meaning of the original data. There are many discretization methods used in the literature, the simplest and most commonly used one is called "equal width" [33] in which the range of the values of the class is divided into equal intervals and then to give each interval a discrete value.

As discussed earlier and as it will be shown in details in the experiments, three parameters are used, namely: duration, latency, and speed. The ranges of continuous values for the three parameters

are divided into 5 intervals with equal lengths with the following discrete values:

For Duration: the discrete values are veryshort, short, normal, long, and verylong, and for Latency: veryshort, short, normal, long, and toolong, while the discrete values for Speed are: veryslow, slow, average, fast, and veryfast. The decision class owner has two values: yes for the owner and no for adversaries. Now, all values are discrete and ready to be used by ILA.

d. Applying ILA to Produce Rules:

ILA takes the training set that contains the entries of the owner and adversaries in discrete form and produces the classification rules in the form IF … THEN …. More details about the implementation of ILA and how it works are shown in the following subsection.

e. Classification

The rules that are produced by ILA can be inserted into the system and used to distinguish the owner from adversaries. These rules should help the system to accept the owner all the time and at the same time rejects adversaries from accessing the system.

4 An Illustrative Example

Let's go along the steps of the system through the following example:

The password that will be used in this example is "hello". The three parameters; as mentioned earlier; are:

- The *duration* of passing each character. We have 5 durations, since there are 5 characters in the password.
- The *latency* between each two consecutive characters. We have 4 latencies: he, el, ll, and lo.
- The *overall time* spent for entering the whole password (this will be called the *speed*).

So, we have 10 attributes for "hello" in addition to a binary decision attribute which indicates that the person entered the password is either the owner of the system or not. These attributes will be as follows: h, e, l, l, o, he, el, ll, lo, speed, and owner.

Now, the owner and the adversaries will enter the password, and then the system will extract the values of these attributes for each entry. A snapshot of 5 entries of the collected values (in milliseconds) is shown in Table 1.

Table 1 A snapshot of five collected entries for the password "hello"

h	e	l	l	o	he	el	ll	lo	speed	owner
33.0019	0	0	1.0001	0	0	0	47.0026	0	4466.256	no
64.0036	80.0046	65.0037	76.0044	57.0032	372.0213	152.0087	245.014	89.0051	3121.179	yes
60.0035	77.0044	71.004	70.004	64.0036	361.0206	148.0085	259.0149	91.0052	3453.198	yes
66.0038	76.0043	69.0039	72.0041	73.0042	362.0207	162.0093	257.0147	89.0051	3710.212	yes
58.0033	72.0041	60.0034	72.0042	56.0033	405.0232	240.0137	264.0151	96.0054	2990.171	no

The values appeared in Table 1 are continuous values. In order to use these values with ILA, they must be discretized using a discretizing method. As discussed earlier, the simplest method and most commonly used one is called "equal width" method. Table 2 shows the discrete values of those appeared in Table 1.

Table 2 The discretized values of five entries for the password "hello"

h	e	l	l	o	he	el	ll	lo	speed	owner
tooshort	tooshort	tooshort	tooshort	tooshort	veryshort	veryshort	veryshort	veryshort	slow	no
short	long	short	normal	veryshort	short	veryshort	short	veryshort	fast	yes
short	normal	normal	normal	short	short	veryshort	short	veryshort	fast	yes
short	normal	short	normal	normal	short	veryshort	short	veryshort	average	yes
veryshort	normal	short	normal	veryshort	normal	short	short	veryshort	veryfast	no

The entries are now ready to be used by ILA. ILA is a powerful algorithm that accepts examples with discrete values and produces the minimum number of rules in a general form with minimum number of conditions. Figure 1 shows the results of applying ILA on 50 examples for the password "hello".

The resulted rules are used to check the entered password by anybody to access the system applying PaSSIL after extracting the same attributes from it, if a rule with decision yes

fires, then the person trying to access the system is its owner and then is granted the permission to access the system, otherwise he will be considered as an adversary and prevented from doing that.

5 Experiments and Results

Many experiments have been conducted to evaluate the system; each experiment is done on a different password with a different length. These passwords are: "houseroad91", "hello", and "arroundtheworldin80days" and entered by hundreds of persons of a diversity of backgrounds and educational levels at different times but on the same keyboard.

```
Data Set File Name: 50 examples for hello.csv
Number of Attributes: 10
Number of Classes  : 2
Number of Examples: 50
Evaluation Method  : Random Sampling
Percentage of Unseen Set  : 20%
Number of experiments  : 1
Number of training samples: 40
Number of unseen samples: 10
=======================================
Experiment # 1
=======================================
Number of rules: 10
Average Number of conditions: 1.9
Rules:
If ll = veryshort => yes
If l = verylong and lo  = short => yes
If h = verylong and e  =verylong and l  = verylong => yes
If l = normal => no
If el = short and speed  = slow => no
If e = normal and speed  = slow => no
If l = toolong and speed  = slow => no
If he = toolong and speed  = slow => no
If e = long and ll  = short => no
If l = short and ll  = short => no
Session results:
Accuracy:  90.909090909091%
Precision:  87.5%
Recall:  93.75%
K-Folds  92.3%
FI Score:  89.523809523809%
Total time Consumed is: 00:00:02.0076883
```

Fig. 1 A snapshot of rules generated by ILA of 50 examples for the password "hello"

In order to evaluate the system thoroughly, four evaluation methods are used in each experiment. These methods are explained briefly as follows:

i. **Hold Out method** in which the data set is divided into two groups: the training set which is used to train the classifier and the test set which is used to estimate the error rate of the trained classifier. This process is a single train-and-test experiment.

ii. **Random Sampling**, in which the dataset is split randomly into a fixed number of examples without replacement. For each data split we retain the classifier from scratch with the training examples and the estimate E_i with the test examples. The true error estimate E is obtained as the average of the separate estimates.

iii. **Leave-One-Out cross validation** in which; for a data set with N examples, N experiments are performed, for each experiment N-1 examples are used as training examples and the remaining example for testing. The true error is estimated as the average error rate on test examples.

iv. **Boot strap method** which is a resampling technique with replacement. From a dataset with N examples, N examples are randomly selected with replacement and used for training; while the remaining examples that are not selected are used for testing. This value is likely to change from fold to fold. This process is repeated for a specified number of folds. The true error is estimated as the average error on test data.

v. **K-Folds cross validation method** in which the data set is divided into k folds, for each k experiments, K-1 folds are used for training and a different fold for testing. The advantage of this method is that all the examples in the training set are eventually used for both training and testing.

In each of the above methods, four measurements are considered for evaluating the system. These measurements are summarized as follows:

1. **Precision** which measures how many of the examples classified as positive are actually positive, and calculated as follows: Precision = TP/(TP+FP)

2. **Recall** which measures how many of the total positive examples were classified as positive, and calculated as follows: Recall = TP/(TP+FN)

3. **F1 Score** which is high only when both the precision and recall are high. This measure is calculated as follows: F1 Score = (2*precision * recall)/(precision + recall)

4. **Accuracy** which measures the accuracy of the algorithm in classifying the test examples, and is calculated as follows: Accuracy = number of correctly classified test examples/total number of test examples

Where:

- **TP** is an abbreviation of True Positives which means owners correctly classified as owners
- **TN** is an abbreviation of True Negatives which means adversaries correctly classified as adversaries
- **FP** is an abbreviation of False Positives which means adversaries misclassified as owners
- **FN** is an abbreviation of False Negatives which means owners misclassified as adversaries

Our goal is to maximize these measurements. Actually, maximizing the precision only or the recall only is easy, but maximizing both is difficult and should be the goal.

The set of experiments that are conducted on the password "houseroad91" are with three data sets each with different number of examples; 150, 210, and 265 examples. These three data sets are described in Table 3. Each experiment is conducted 5 times, and the average of these five times is calculated and considered for the evaluation. The data in these sets and other data sets in other experiments were collected by hundreds of persons with many entries of the same person for the same password, in different times with different moods and atmospheres. Part of the data set was entered by one person as the owner of system and the rest are entered by different persons as adversaries. To study the effect of the size of the data set on the accuracy of the system, the same owner is considered in the three data sets.

Table 3 The description of the three data sets of the password "houseroad91"

	Data set 1	data set 2	data set 3
Total Number of Examples	150	210	265
Number of Examples for Owner	50	50	105
Number of Examples for Adversaries	100	160	160

The number of attributes for this password is 22 plus the decision attribute: 11 attributes for the duration of each character, 10 attributes for the latency of each two consecutive characters, and 1 attribute for the overall speed of entering the password, in addition to the decision attribute.

Figure 2 shows the results of four experiments of applying random sampling on the small data set i.e. 150 examples, with different percentages of unseen (test) examples; i.e. 20%, 33.3%, 50%, and 70%, for the four evaluating methods; namely; accuracy, precision, recall, and F1 score. As noted in the table, for 20% unseen examples we got around 87% accuracy as the average of the results of the four evaluating methods. Despite the fact that in inductive learning, it is known that the results sometimes depend and affected by the nature of the data set and the randomness degree of selecting the examples for testing or for training, however, the accuracy usually decreases as the percentage of unseen examples gets higher. Also it is noted from the table that the accuracy is around 82%, 84%, and 80% for 33.3%, 50%, and 70% unseen examples respectively. This is good, especially the accuracy obtained for 70% unseen examples for the smallest data set with 150 examples.

Figure 3 shows the results of experiments similar to the above, but using the other three evaluation methods, namely; leave-one-out, hold out, boot strapping, and K-Folds. The results obtained for these three methods are almost close to each other for the four measurements, where they got 92%, 81%, 87% and 86% respectively.

Table 4 shows the exact values of the results of the above experiments.

The above results are obtained on a relatively small data set. Let's now repeat the same experiments, but on a larger data set with 210 examples.

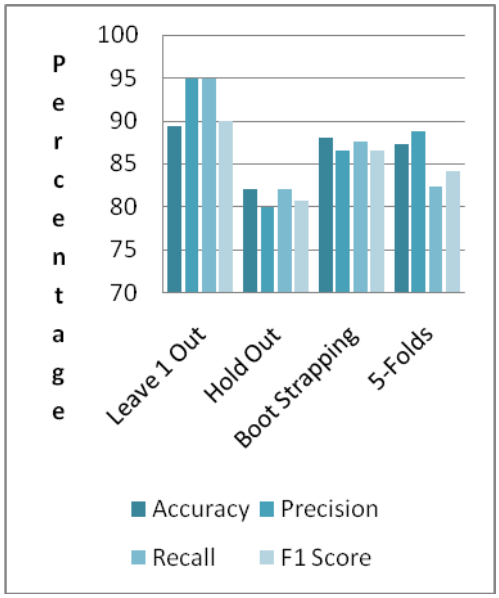

Fig. 2 Random Sampling on 150 examples of the password "houseroad91"

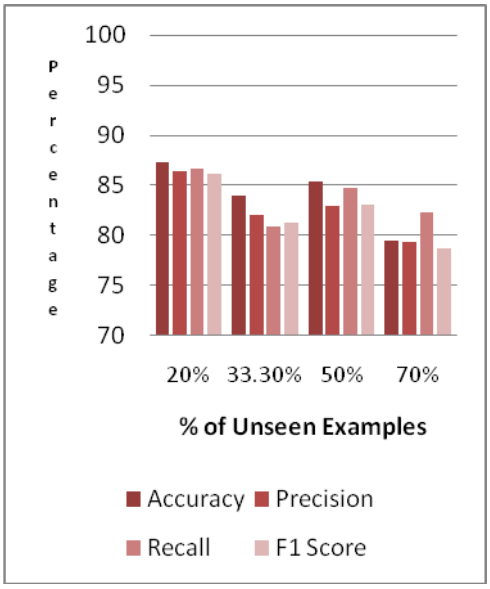

Fig. 3 Other evaluation methods on 150 examples of the password "houseroad91"

In this data set, entries of the owner remain the same, i.e. 50 entries, while the entries of the adversaries increased by 60 entries to be 160 entries. Actually, it is expected to get good results since increasing adversaries' entries enriches the data set with new patterns for new users and hence increases the induction ability of the system.

Figures 4 and 5 depict the results of this experiment. A slight improvement has been obtained regarding the random sampling, while almost the same results have been obtained for the other evaluation methods.

This means that the strength of the inference ability of ILA has been improved but the quality of the system inference remains the same. This is normal because any inductive algorithm increases its ability to infer as number of examples gets higher and higher, but this increase is not necessary to improve the quality of the results unless the added examples are chosen carefully. It seems that increasing adversaries' entries is not enough, so as depicted in Table 3, let's increase the owner entries by 55 to become 105 and that for adversaries to remain as 160, and observe what will happen.

Table 4 Results of all experiments on the data set of 150 examples of the password "houseroad91"

	Random Sampling				Leave One Out	Hold Out	Boot Strapping	K-Folds
	20% unseen	33.30% unseen	50% unseen	70% unseen				
Accuracy	87.3	84	85.3	79.43	89.333	82.0	88.0	87.3
Precision	86.4	82.0	83.0	79.4	95.0	80.0	86.6	88.8
Recall	86.7	80.9	84.7	82.3	95.0	82.1	87.6	82.3
F1 Score	86.2	81.3	83.1	78.7	90.0	80.7	86.6	84.1

As depicted in Figures 6 and 7, significant improvements have been obtained in both the inference ability of the algorithm and the quality of the results as well. This may be justified by the fact that increasing number of the owner entries helps in well recognizing the pattern of the user, since he/she entered the password more times and in more situations and different moods and atmospheres, while; as stated earlier, increasing the adversaries' entries helps in recognizing more patterns for adversaries which excludes these patterns from the owner patterns.

Another experiment has been conducted on a shorter password with 5 characters length; namely "hello". To compare the results of this experiment

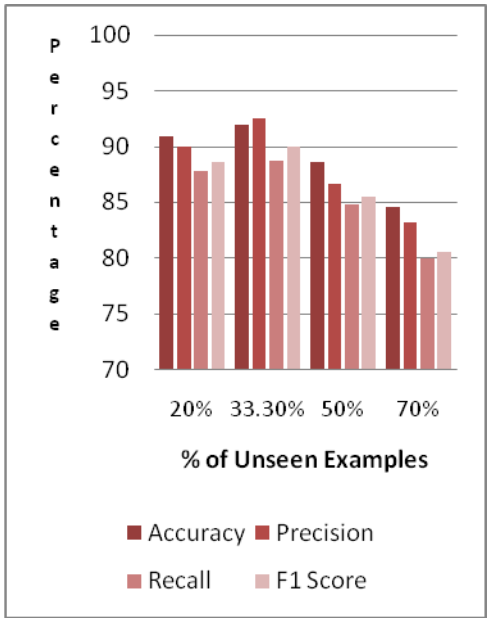

Fig. 4 Random Sampling on 210 examples of the password "houseroad91"

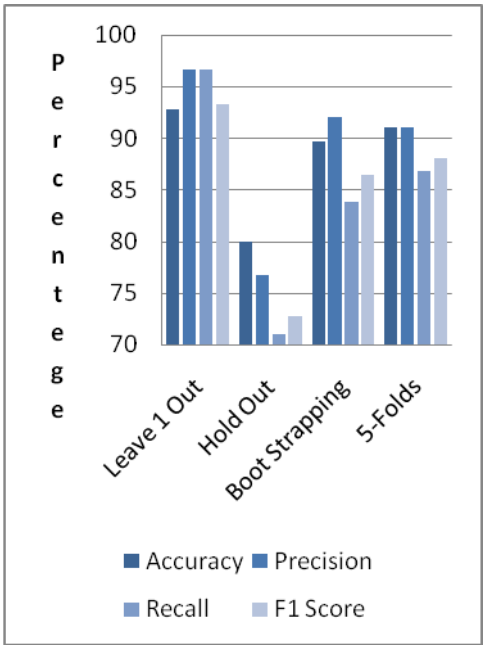

Fig. 5 Other evaluation methods on 210 examples of the password "houseroad91"

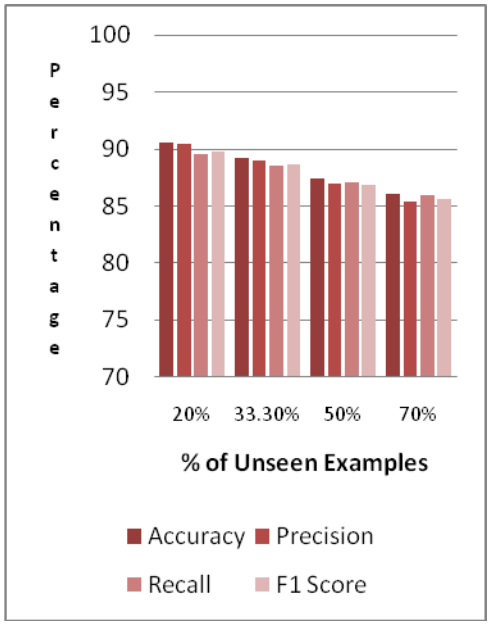

Fig. 6 Random Sampling on 265 examples of the password "houseroad91"

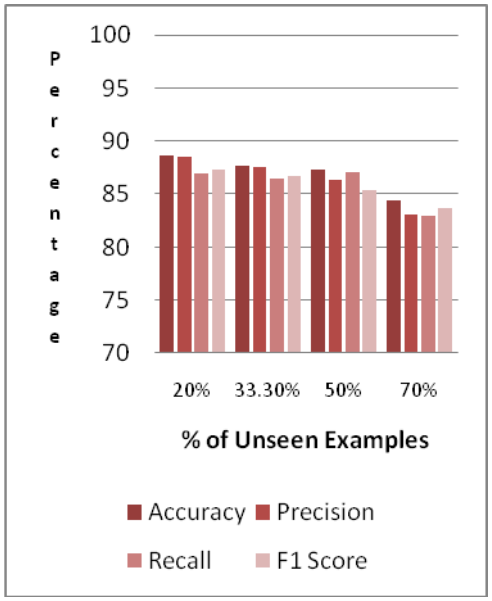

Fig. 8 Random Sampling on 265 examples of the password "hello"

with the previous one, the same sample that entered the password "houseroad91" with 265 examples the password "houseroad91". This is normal, because as the password gets shorter and shorter; also entered the password "hello"; with 60 for the owner entries and 105 entries for adversaries.

Figures 8 and 9 show the results of the system for the password "hello". As noted from the figures, the accuracy of the system has been slightly degraded when compared with the case of entering the same pattern is more likely to be similar among the owner and adversaries.

In the last experiment, we will examine the system with a longer password with 23 characters, namely; "arroundtheworldin80days". To make the comparison real, this password was entered by the same persons in data set 3. It is expected to get good results, since the password is long. With long passwords, the opportunity to have similar patterns among users is very rare.

Figures 10 and 11 depict the results of this experiment. As seen in the figures below, the results are better that what have been obtained from the previous experiments.

Figures 12 to 14 show a comparison between PaSSIL and the two statistical methods DLO [6] and DLOS [7] for the three passwords; "hello", "houseroad91", and "aroundtheworldin80days" respectively. PaSSIL shows better results in most cases compared with DLO and DLOS. In addition, PaSSIL is an inductive learning system based on keystroke dynamics while DLO and DLOS are statistical static methods that are based on a series of statistical calculations. This means that PaSSIL overcomes the gradual changes in the way the owner enters the password while DLO and DLOS do not.

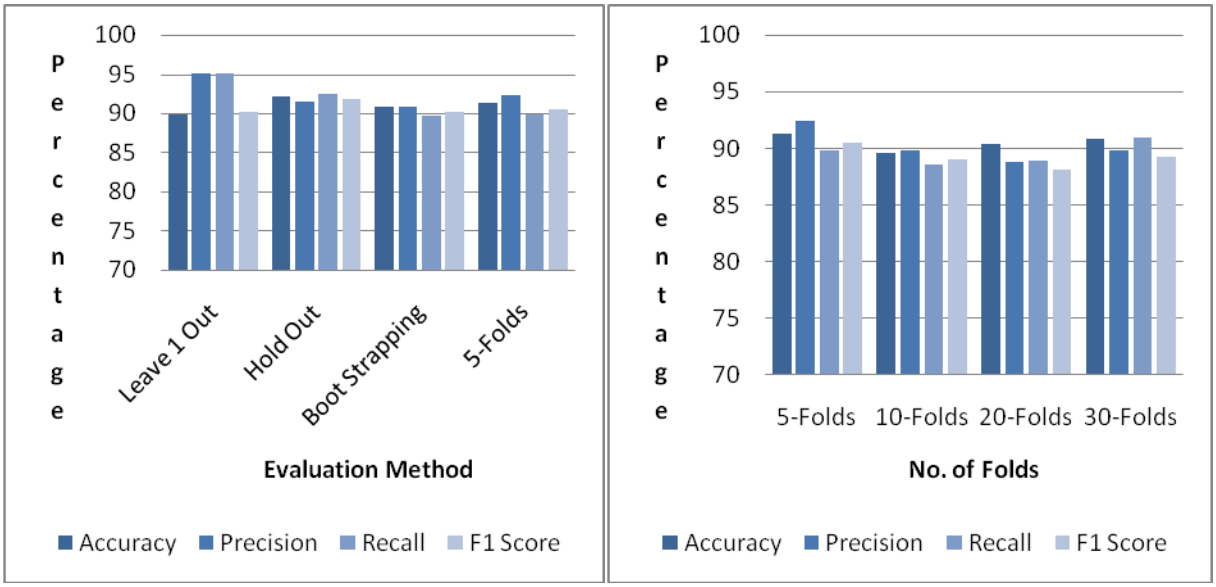

Fig. 7 Other evaluation methods on 265 examples of the password "houseroad91"

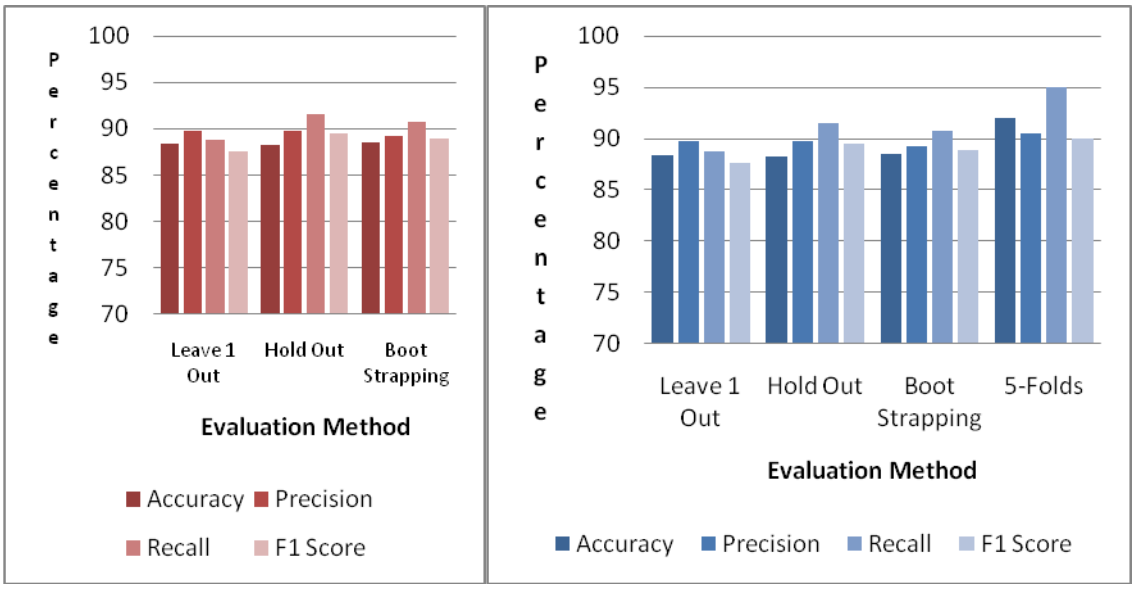

Fig. 9 Other evaluation methods on 265 examples of the password "hello"

To complete the whole picture, our work is compared with some other known approaches in this area, even though there are a huge number of algorithms and approaches for tackling the password security issue through keystroke biometrics, each approach is with different evaluation method, different number of parameters, different number of password entries needed, and different number of attributes. This diversity makes it difficult to compare one approach with another. The most suitable way is to compare algorithms with the same group and the same dataset.

For this purpose, we will compare our work with three known machine learning approaches, i.e. the standard backpropagation algorithm with 3 layers [22], a modified version of PNN algorithm [35],

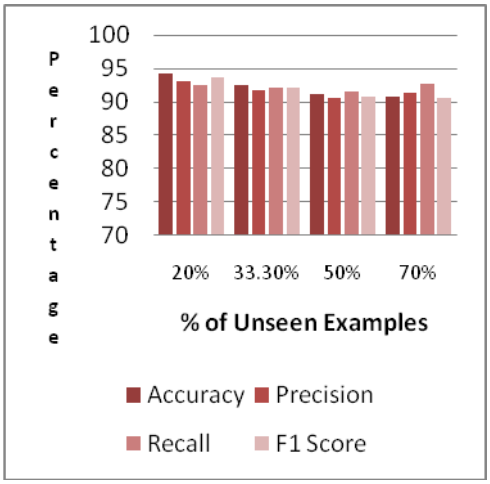

Fig. 10 Random sampling on 265 examples of the password "arroundtheworldin80days"

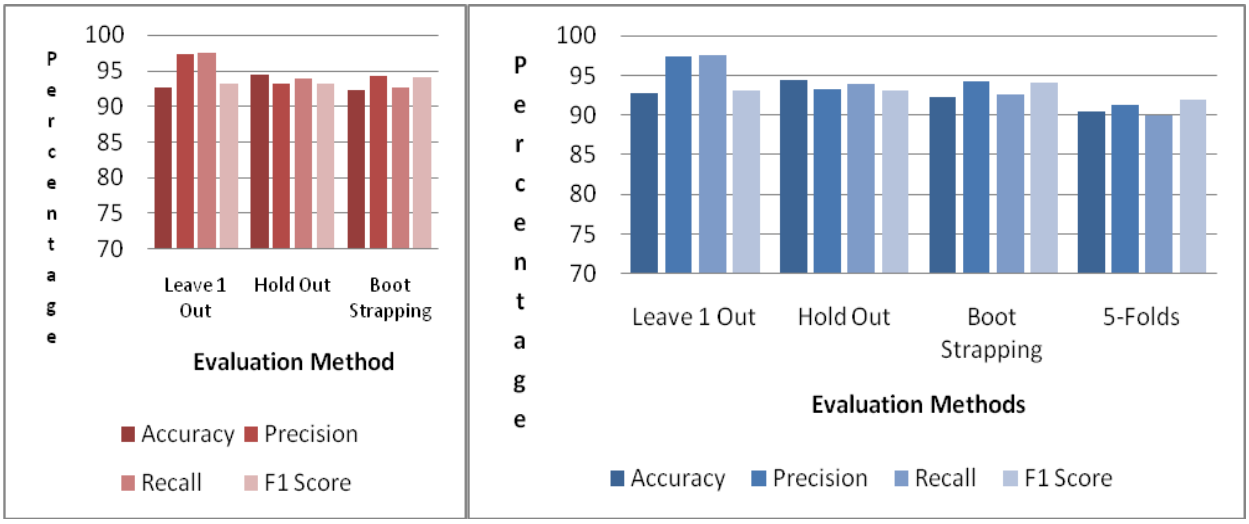

Fig. 11 Other evaluation methods on 265 examples of the password "arroundtheworldin80days"

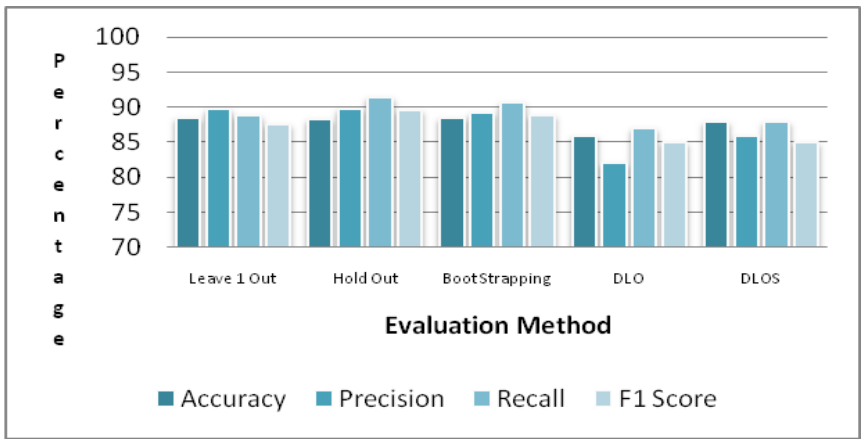

Fig. 12 Comparison between PaSSIL (the first three columns), DLO, and DLOS for the password "hello"

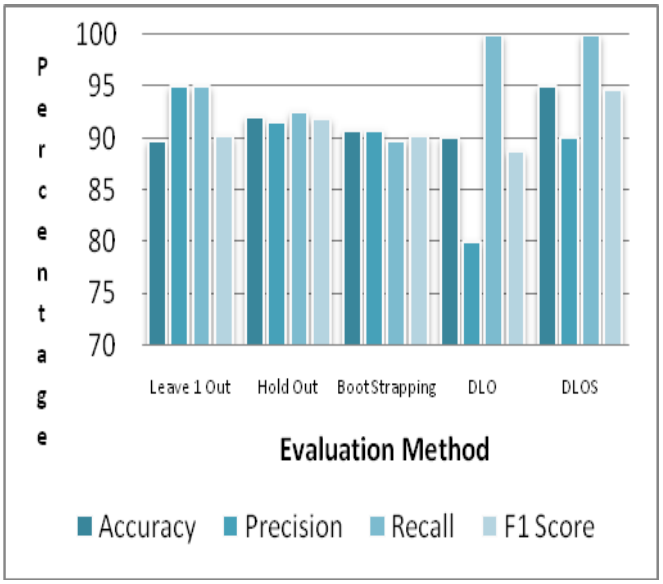

Fig. 13 Comparison between PaSSIL (the first three columns), DLO, and DLOS for the password "houseroad91"

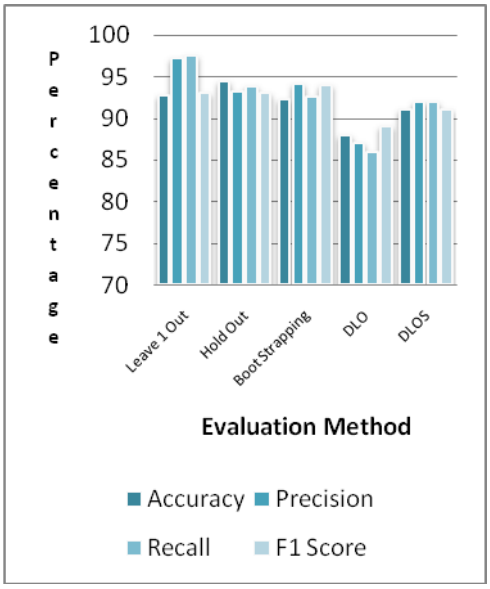

Fig. 14 Comparison between PaSSIL (the first three columns), DLO, and DLOS for the password "aroundtheworldin80days"

and the Quinlan's ID3 algorithm [36]. To accomplish this task, a data set of 250 samples for the password "hellonewworld123"has been collected from the valid user and 50 adversaries. Random subsets of 50% of this sample have been used for training and 50% is for testing. The experiments have been repeated 10 times and to ensure that all examples are tested, it has been evaluated with k-folds cross evaluation method with 10-folds, and then the average accuracy has been considered. For the backpropagation and ID3 algorithms, the same attributes of our work have been used. But for the modified PNN algorithm, we preferred to use the same attributes they used in their work with which their algorithm produced the best results. Table 5 shows the accuracy values for recognizing legitimate users for different sizes of data sets. It is worthy to say that the values obtained are data set dependent and change from one experiment to another, but these values are the average accuracy for 10 trials of each training set. As it is noted from the table, PaSSIL got better results in most cases.

Table 5. The results of comparing PaSSIL with BP, ID3, and modified PNN algorithms

Training Set Size	ID3	BP	Modified PNN	PaSSIL
100	91.62%	91.52%	95.05%	96.13%
150	92.42%	91.90%	96.64%	96.50%
200	93.37%	92.37%	96.90%	97.04%

The experiments showed that the computational training time of PaSSIL is better than that of all other algorithms, since for the 200 examples, ID3 took around 2 min in the training, BP took approximately 3.6 min and modified PNN took 53 sec while PaSSIL took less than 32 sec. And for the classification time, PaSSIL is slightly lower than others since it took around 3 sec while others took around 5 sec on average.

6 Conclusions

PaSSIL is a new Inductive Learning System for Password Strengthening Based on Keystroke Dynamics. This system takes into account the gradual change in the pattern of the owner of the system that is caused by changing the moods, atmospheres, situations, and times of entering the password.

PaSSIL had been tested by many experiments on hundreds of users with different educational levels and ages. The results showed that PaSSIL is a powerful system and comparable with other systems if not better. PaSSIL can be easily embedded and used in any system.

PaSSIL is a simple system that extracted three parameters from the entered password, namely duration, latency, and speed. With these three parameters, it obtained above 90% accuracy in all experiments done. It can be further improved by introducing more parameters, but this must be done

carefully so as not to affect the complexity of the system.

The main drawback of PaSSIL is that all experiments must be conducted on a certain password. If the owner decides to change the password, the whole experiments should be repeated on the new password. We are now working on a system that works regardless of certain passwords and without conducting experiments on adversaries, in which, the owner can change his password whenever he wants without changing the system.

Acknowledgements

Many thanks go to Dr. Sufyan Al Majali for programming ILA and to Dr. Ibrahim Al Bluwi for his valuable contributions on evaluation methods. Many thanks also go to my student Hassan Nuri for programming the data collecting and parameterization system, and to all the volunteers who helped us with entering the data needed for the experiments.

7 References

[1] T. Sim and R. Janakiraman. Are digraphs good for free-text keystroke dynamics? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

[2] R. Gaines, W. Lisowski, S. Press, and N. Shapiro. Authentication by keystroke timing: some preliminary results. Technical Report Rand Rep. R-2560-NSF, RAND Corporation, 1980.

[3] F. Monrose and A.D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation Computing Systems*, 16(4):351–359, 2000.

[4] R. Giot, M. El-Abed, and C. Rosenberger. Web-based benchmark for keystroke dynamics biometric systems: a statistical analysis. In *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, pages 11–15, 2012.

[5] D. Hosseinzadeh and S. Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(6):816–826, 2008.

[6] Abu-Soud S., A Hybrid Key Stroke Authentication System. The Proceeding of the Second International Conference on Informatics Engineering & Information Science (ICIEIS2013), Malaysia. Nov. 12-14, 2013. pp 84-95.

[7] Abu-Soud S. "An Enhanced Non-Static Biometric Keystroke Dynamics Model for Strengthening the Information Content Security", International Journal of Applied Science and Technology, Vol. 4, No. 5; September 2014, 219-232.

[8] S. Bleha, C. Slivinsky, and B. Hussien. Computer-access security systems using keystroke dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1217–1222, 1990.

[9] M. Brown and S.J. Rogers. User identification via keystroke characteristics of typed names using neural networks. *International Journal of Man-Machine Studies*, 30(6):999–1014, 1993.

[10] J.D. Allen. An analysis of pressure-based keystroke dynamics algorithms. Master's thesis, Southern Methodist University, Dallas, TX, U.S.A., 2010.

[11] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176, 1990.

[12] P. Kang, S. Hwang, and S. Cho. Continual retraining of keystroke dynamics based authenticator. In *International Conference on Advances in Biometrics (ICB)*, volume 4642 of *LNCS*, pages 1203–1211, 2007.

[13] S. Haider, A. Abbas, and A.K. Zaidi. A multi-technique approach for user identification through keystroke dynamics. In *IEEE International Conference on Systems, Man, and Cybernetics (ICSMC)*, volume 2, pages 1336–1341, 2000.

[14] S. Cho, C. Han, D.H. Han, and H.I. Kim. Web-based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10(4):295–307, 2000.

[15] R.S. Zack, C.C. Tappert, and S.H. Cha. Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method. In *4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–6, 2010.

[16] N. Bartlow and B. Cukic. Evaluating the reliability of credential hardening through keystroke dynamics. In *17th International Symposium on Software Reliability Engineering (ISSRE)*, pages 117–126, 2006.

[17] E. Yu and S. Cho. GA-SVM wrapper approach for feature subset selection in

keystroke dynamics identity verification. In *International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 2253–2257, 2003.

[18] W. Martono, H. Ali, and M.J.E. Salami. Keystroke pressure-based typing biometrics authentication system using support vector machines. In *International Conference on Computational Science and Its Applications (ICCSA)*, volume 4706 of *LNCS*, pages 85–93, 2007.

[19] Y. Deng and Y. Zhong. Keystroke dynamics user authentication based on Gaussian mixture model and deep belief nets. *ISRN Signal Processing*, 2013, Article ID 565183, 7 pages, 2013.

[20] J. Leggett, G. Williams, M. Usinick, and M. Longnecker. Dynamic identity verification via keystroke characteristics. *International Journal of Man-Machine Studies*, 35(6):859–870, 1991.

[21] D.T. Li. Computer-access authentication with neural network based keystroke identity verification. In *International Conference on Neural Networks*, volume 1, pages 174–178, 1997.

[22] Bleha, S. A., Knopp, J. and Obaidat, M.S. Performance of the perceptron algorithm for the classification of computer users, Proceedings of the ACM/SIGAPP Symposium on Applied Computing, New York Press, 2002.

[23] C.C. Loy, C.P. Lim, and W.K. Lai. Pressure-based typing biometrics user authentication using the fuzzy ARTMAP neural network. In *International Conference on Neural Information Processing (ICONIP)*, 2005.

[24] C.C. Loy, W.K. Lai, and C.P. Lim. Keystroke patterns classification using the 20 *Y. Zhong and Y. Deng* ARTMAP-FD neural network. In *3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, volume 1, pages 61–64, 2007.

[25] R. Bixler and S. D'Mello. Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. In *International Conference on Intelligent User Interfaces (IUI)*, pages 225–234, 2013.

[26] C. Epp, M. Lippold, and R.L. Mandryk. Identifying emotional states using keystroke dynamics. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 715–724, 2011.

[27] Tolun M. and Abu-Soud S., "An Inductive Learning Algorithm for Production Rule Discovery," The International Journal of Expert Systems with Applications, 14(3), April 1998, 361-370.

[28] Abu-Soud S., "A Framework for Integrating Decision Support Systems and Expert Systems with Machine Learning", Proceeding of the 10th International Conference on Industrial and Engineering Applications of AI and ES, June 1997, Atlanta, USA.

[29] Abu-Soud S., "A Disjunctive Learning Algorithm for Extracting General Rules", Journal of Institute of Mathematics and Computer Science (Computer Science Series), Vol. 10, No. 2 (1999) 201-217.

[30] Haj Hassan M. and Abu-Soud S., "A Parallel Inductive Learning Algorithm," AMSE journal, France, Dec. 2000.

[31] Oludag M., TounM.,Sever , and Abu-Soud S., "ILA-2: An Inductive Learning Algorithm for Knowledge Discovery.", Cybernetics and Systems: An International Journal, vol. 30, no. 7, Oct.-Nov. 1999.

[32] Abu-Soud S. and Al Ibrahim A., DRILA: A Distributed Relational Inductive Learning Algorithm, WSEAS Transactions on Computers, Issue 6, Volume 8, June 2009, ISSN: 1109-2750.

[33] D. Chiu, A. Wong, and B. Cheung, "Information Discovery through Hierarchical Maximum Entropy Discretization and Synthesis", Knowledge Discovery in Databases, G. Piatesky-Shapiro and W.J. Frowley, ed., MIT Press, 1991.

[34] F. Gorunescu et al. A cancer diagnosis system based on rough sets and probabilistic neural networks, 5th European Conference on Health care Modelling and Computation, University of Medicine and Pharmacy of Craiova, pp. 149-159.

[35] K. Revett et al. A machine learning approach to keystroke dynamics based user authentication, Int. J. Electronic Security and Digital Forensics, Vol. 1, No. 1, 2007.

[36]Quinlan, J.R.(1983). "Learning Efficient Classification Procedures and their Application to Chess End Games". In R.S. Michalski, J.G. Carbonell& T.M. Mitchell, Machine Learning, an Artificial Intelligence Approach, Palo Alto, CA: Tioga, 463-482.