CHAPTER 6

PARTICLE SWARM OPTIMIZATION AND
GENETIC ALGORITHM IN RELATION TO
KEYSTROKE DYNAMICS

6.1 INTRODUCTION

Evolutionary computational algorithms have been the preoccupation of the researchers and professionals in the domain of computer security in recent times. Genetic Algorithms and Particle Swarm Optimization are the trendiest approaches that are concentrated upon in the optimization problems. Since these two approaches are tried in finding a solution to a given objective function with the employment of different strategies and computational effort, it is germane to assess their relative values. In fact, the mainstay of this thesis, the keystroke dynamics, a rather promising technique in recent times, is an area where feature selection is done using a stochastic optimization technique, Particle Swarm Optimization. It is essential to know which biometric technique – Genetic Algorithm, a well-established yet older version or Particle Swarm Optimization, a recent technique among many employed in continuous optimization problems - is better. The two key performance indicators frequently used for comparison are the solution quality and solution time. A general description of the algorithms at the conceptual level is followed by a discussion of their virtual advantages in keystroke dynamics.
6.2 EVOLUTIONARY APPROACH

In all evolutionary algorithms, there are basically three processes. The first process is the initialization process where the initial process of individuals is randomly generated according to some solution representation. Each individual represents a solution, directly or indirectly. If an indirect representation is used, each individual must first be decoded into a solution. Each solution in the population is then evaluated for fitness value in the second process. The third process is the generation of a new population by perturbation of solutions in the existing population. The key processes are represented in the Figure 6.1.

![Flow chart of evolutionary algorithm](image)

**Figure 6.1 Flow chart of evolutionary algorithm**

In making use of the evolutionary algorithm in solving optimization problems, the first important task is to determine how the solution can be represented according to the elements or terminology of the specific
evolutionary algorithm. The processes for initialization and generation of new population may produce infeasible solutions. It is very important to choose a solution representation that is more likely to produce feasible solutions. This is a common design consideration for all evolutionary algorithms. The solution representation can be a direct or indirect one. The main design consideration is to ensure that each individual generated can always be decoded into a feasible solution. For a complex problem, indirect representation is often used along with a decoding procedure to convert the indirect solution representation into a feasible solution. Once the solution is decoded, the fitness function can be evaluated. In addition to the solution representation, two general parameters which should be determined initially are the population size and the maximum number of iteration. The choices of values of these two parameters have major influence on the solution quality and solution time, and in practice, these values are almost always determined empirically through pilot runs.

6.3 GENETIC ALGORITHM

Even though GA’s origins are much earlier than 1975, Holland’s (1975) seminal thesis in that year made a widespread influence on all horizons. The main concept of GA is to mimic the natural selection and survival of the fittest theories enunciated in genetic engineering. The Figure 6.2 illustrates the genetic algorithm.
In order to understand the working process of genetic algorithm, the principal phases in the realization of GA are reproduced (Jones & Karl O Jones 2005).

- **Step 1**: Represent the problem variable domain as a chromosome of fixed length; choose the size of the chromosome population $N$, the crossover probability $P_c$ and the mutation probability $P_m$.

- **Step 2**: Define a fitness function to measure the performance of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

- **Step 3**: Randomly generate an initial population of size $N$: $x_1$, $x_2$, $x_N$. 

---

**Figure 6.2 Flowchart of genetic algorithm**

![Flowchart of genetic algorithm](image)
- **Step 4:** Calculate the fitness of each individual chromosome:
  \( f_1, f_2, ..., f_N \)

- **Step 5:** Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. High fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

- **Step 6:** Create a pair of offspring chromosomes by applying the genetic operators.

- **Step 7:** Place the created offspring chromosomes in the new population.

- **Step 8:** Repeat Step 5 until the size of the new population equals that of initial population, \( N \).

- **Step 9:** Replace the initial (parent) chromosome population with the new (offspring) population.

- **Step 10:** Go to Step 4, and repeat the process until the termination criterion is satisfied.

A GA is an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500. A common practice is to terminate a GA after a specified number of generations and then examine the best chromosomes in the population. If no satisfactory solution is found, then the GA is restarted (Koza 1991).

### 6.4 PARTICLE SWARM OPTIMIZATION

PSO is a variety in the massive category of Swarm Intelligence Methods. Kennedy originally proposed PSO as a simulation of social behaviour and its introduction as an optimization method took place in 1995.
(Kennedy & Eberhart 1995). PSO is easier in implementation and computationally inexpensive compared to other genetic algorithms. The Figure 6.3 represents the Particle Swarm Algorithm.

**Figure 6.3 Flowchart of Particle Swarm Algorithm**

PSO optimises an objective function by undertaking a population-based search. The population consists of potential solutions, named particles, which are a metaphor of birds in flocks. These particles are randomly initialised and freely fly across the multidimensional search space. During flight, each particle updates its own velocity and position based on the best experience of its own and the entire population. The updating policy drives the particle swarm to move toward the region with the higher objective function value, and eventually all particles will gather around the point with
the highest objective value. The detailed operation of particle swarm optimization is as follows:

Step 1: **Initialisation.** The velocity and position of all particles are randomly set to within pre-defined ranges.

Step 2: **Velocity Updating.** At each iteration, the velocities of all particles are updated according to:

\[
\vec{V}_i = w \vec{V}_i + c_1 R_1 (\vec{P}_{i,\text{best}} - \vec{P}_i) + c_2 R_2 (\vec{g}_{i,\text{best}} - \vec{P}_i)
\]

where \( \vec{P}_i \) and \( \vec{V}_i \) are the Position and velocity of particle \( i \), respectively \( \vec{P}_{i,\text{best}} \) and \( \vec{g}_{i,\text{best}} \) is the position with the ‘best’ objective value found so far by particle \( i \) and the entire population respectively; \( w \) is a parameter controlling the flying dynamics;

\( R_1 \) and \( R_2 \) are random variables in the range \([0, 1]\); \( c_1 \) and \( c_2 \) are factors controlling the related weighting of corresponding terms. The inclusion of random variables endows the PSO with the ability of stochastic searching. The weighting factors, \( c_1 \) and \( c_2 \), compromise the inevitable trade off between exploration and exploitation. After updating, \( i \) \( v \) \( r \) should be checked and secured within a pre-specified range to avoid violent random walking.

Step 3: **Position Updating.** Assuming a unit time interval between successive iterations, the positions of all particles are updated according to

\[
\vec{P}_i = \vec{P}_i + \vec{V}_i
\]

After updating, \( \vec{P}_i \) should be checked and limited to the allowed range.
Step 4: Memory updating. Update $i$, best $p$ rand $i$ best $g$, $r$ when condition is met.

$$\vec{P}_{i,\text{best}} = \vec{P}_i \quad \text{if} \quad f(\vec{P}_i) > f(\vec{P}_{i,\text{best}})$$

$$\vec{g}_{i,\text{best}} = \vec{g}_i \quad \text{if} \quad f(\vec{g}_i) > f(\vec{g}_{i,\text{best}})$$

where $f(\vec{x})$ is the objective function subject to maximization.

Step 5: Termination Checking. The algorithm repeats Steps 2 to 4 until certain termination conditions are met, such as a pre-defined number of iterations or a failure to make progress for a certain number of iterations. Once terminated, the algorithm reports the values of $\vec{g}_{\text{best}}$ and $f(\vec{g}_{\text{best}})$ as its solution.

PSO algorithm does not require sorting of fitness values of solutions in any process. This might be a significant computational advantage over GA, especially when the population size is large. Also the updates of velocity and position in PSO require a simple arithmetic operation of real numbers.

6.5 KEYSTROKE DYNAMICS

Keystroke dynamics is a biometric based on the assumption that people type in an inimitable characteristic manner. Keystroke dynamics is mostly applicable to verification, besides identification possibility. In verification, it is recognized who the user is supposed to be and the biometric system should verify if the user is who he claims to be. In identification, the biometric system should identify the user without any additional knowledge, using only keystroke dynamics. Most applications of keystroke dynamics are in the field of verification. Keystroke analysis is essentially a form of Pattern Recognition, as are, in fact, most other biometric techniques. By itself, it
involves representation of input data measures, extraction of characteristic features and classification or identification of patterns data in an attempt to decide on to which pattern class these data belong. This authentication system consists of three steps:

i) Feature Extraction

ii) Preprocessing

iii) Feature Subset Selection (GA)

6.5.1 Feature Extraction

The input data is usually represented by a sequence of typed keys, together with appropriate timing information expressing the exact time at which keys have been depressed and released. From the input data, some of the features will be extracted.

i) Duration of typed keys (how long a key is held down)

ii) Latency between two consecutive keystrokes (the elapsed time between the release of the first key and the depression of the second key)

iii) Digraph

iv) Tri-graph

v) Pressure of keystroke

vi) Force of keystroke.

All the features are not constructive and extensively used. For measuring the pressure and the force of keystroke, a special type of pressure or force sensitive keyboard is indispensable. Technical hitches of typing a text, frequency of errors and typing rate are effective with regard to a long
text. The timing features such as Duration or Dwell time, Latency or Flight time, Digraph and Tri-graph are recurrently measured from keystroke. Digraphs and Tri-graphs are sequences of two and three characters correspondingly. Tri-graphs might also be used for some EBCDIC code pages that lack characters such as {and}. In addition to these timing features, a new feature labelled Virtual Key Force is also introduced.

The Virtual Key Force (VKF) is calculated derived from the typing speed and the behaviour of the user during typing using the key board. It measures the time taken by the user between releasing one key and pressing another key. It is founded on the fact that each user has different typing speed and each user takes his individual time to release one key and press another key. The use of keys and the typing speed and force are unlike for discrete users. Also the time interval taken for the release of one key and pressing of another key is not the same for all users. Consequently, virtual key force can be determined from the key complexity.

6.5.2 Preprocessing

The extracted feature may consist of a great deal of superfluous information. Consider the multimodal human verification system that utilizes the combination approach to fusion at the match score level. Min–max and z-score normalization are some of the prevalent techniques used for relevance score normalization in meta search. The most frequently used score normalization technique is the z-score that is calculated using the arithmetic mean and standard deviation of the given data. Given a set of matching scores \( \{n_i\}_{i=1,2,\ldots,n} \), the normalized scores are given by
where $\mu$ is the arithmetic mean and $\sigma$ is the standard deviation of the given data. Pre-processed results are carried over to the next step termed feature subset selection.

### 6.5.3 Feature Subset Selection

Thus a reduced set of genes are got hold of in the previous pre-processing stage. At this second stage, a **wrapper approach** (Figure 6.4) is used that combines a GA and an SVM to accomplish the feature subset selection. Feature subset selection is applied to high dimensional data before proceeding to the classification step. The basic idea is the dependence on using a GA to discover “good” subsets of genes and the reliability of a subset being evaluated by an SVM classifier on a set of training data. In the course of this stage, high quality gene subsets are recorded in an archive for further analysis. At the end of the GA, the exploration of the archived gene subsets is performed: gene subsets are evaluated by putting them side by side and the most recurrently appearing genes are recognized.

\[
 n_i' = \frac{n_i - \mu}{\sigma}
\]

---

**Figure 6.4 Flow Diagram of GA wrapper based feature subset selection**
Contrary to the ordinary GA, GA wrapper has to find not only good but also diverse strings. In order to enforce diversity, the fitness function needs a diversity term as in GEFS. In the current project, one step approach is embraced resembling that of GEFS; yet with a more direct diversity term in the fitness function and SVM as base classifier. The "uniqueness" term is implemented here which measures each chromosome in terms of difference from other chromosomes. As more chromosomes with exclusive features are preferred, distinctiveness is straightforwardly added to accuracy just as diversity term in GEFS.

Support Vector Machines are basically binary classification algorithms. SVM computes the hyper plane that maximizes the margin between the training examples and the class boundary when the data is linearly separable. When the data is not linearly divisible, the examples are mapped to a high dimensional space where such an unscrambling hyper plane can be found. In the wrapper GA/SVM algorithm, an SVM classifier is applied to gauge the quality of a gene subset. The SVM-based classifier is used to ensure the fitness evaluation of each candidate gene subset. One important feature of the GA developed in this composition is the use of an archive to record quality gene subsets discovered during the gene subset selection stage. This archive is then analyzed to identify a small number of decidedly repeatedly appearing genes that are used in the final classification stage.

6.5.4 Outcome Estimation

The human authentication making use of keystroke dynamics is put into operation using MATLAB and it is comprehended that the anticipated practice is much better than other human authentication using keystroke dynamics. In this scheme, an efficient Genetic Algorithm is used for feature reduction. 5×10 chromosomes are engaged in initial population and again
5×10 chromosomes are taken up for second iteration for each person. In this analysis, features of 30 persons are referenced for recognition. As a result, 10×10 total chromosomes are taken into account for each person. The total number of best chromosomes selected is 1500. The total unique index of best chromosomes is 485. By means of these data, the performance of the proposed system is explored.

6.6 PERFORMANCE ANALYSIS

In the study of the performance of the intended process of human authentication system, two terms are taken for deliberation.

- Feature reduction rate
- Error authentication rate

6.6.1 Feature Reduction Rate

Feature selection is also acknowledged as variable selection or feature reduction or attributes selection or variable subset selection. This is the technique of opting for a subset of relevant features for building robust learning models. Subset selection searches the set of possible features for the optimal subset. The feature reduction rate can be calculated for the rate of reduction of the total features. This rate of reduction can be calculated with the help of the total number of features and the number of features selected for authentication. The feature reduction rate can be calculated using the following formula.

\[
\text{Feature reduction} = \frac{\text{Total no. of features} - \text{no. of features selected}}{\text{Total no. of features}}
\]
A comparison of the planned methodology (Table 6.1 and Figure 6.5) is discussed in tandem with three other practices to dissect their qualified performance. They are:

- Genetic Algorithm (GA)
- Particle Swarm Optimization (PSO)
- Ant Colony Algorithm (ACO) with Extreme Machine Learning (ELM)

Extreme Machine Learning (ELM) technique is used as an objective function in GA, PSO and ACO for feature selection. Particle swarm optimization (PSO) is a population dependent stochastic optimization approach. The system is initialized with a population of arbitrary solutions and optima are looked for through updation of generations. The possible solutions in PSO are called particles. The entire quantity of particles follows their coordinates in the problem situation which are related with the best possible solutions (fitness). An additional “best” value that is tracked by the particle swarm optimizer as the best value is achieved at any point by any particle in the neighbourhood of the particle and this location is called Pbest. If a particle considers all the population as its topological neighbours, then the best value is a global best and is called Gbest.

Genetic Algorithm is employed as a computer model in which a population of conceptual representations (called chromosomes) of candidate solutions (called individuals, creatures or phenotypes) to an optimization complexity progresses near better solutions. Fitness function of the GA is determined by the ELM.

The Ant Colony Optimization (ACO) technique has been inspired by the investigation of real ant colony’s foraging activities and on the
principle that ants can frequently identify the shortest path between the food source and the nest. An ant identifying an already laid trail can identify the thickness of pheromone trail. It chooses with a very high probability the shortest path to be followed and strengthens that trail with its own pheromone. If a sizeable quantity of pheromone is on a certain path, the supreme prospect is that an ant chooses that path and for that reason, the path’s pheromone trail will become stronger. Ultimately, the ant colony in concert plots the shortest path that has the highest pheromone density.

Table 6.1 Comparison of Reduction Rate

<table>
<thead>
<tr>
<th>Method</th>
<th>Total no. of Features</th>
<th>No. of Features Selected</th>
<th>Feature Reduction Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA with ELM</td>
<td>43</td>
<td>28</td>
<td>34.88</td>
</tr>
<tr>
<td>PSO with ELM</td>
<td>43</td>
<td>30</td>
<td>30.23</td>
</tr>
<tr>
<td>ACO with ELM</td>
<td>43</td>
<td>23</td>
<td>46.51</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>2550</td>
<td>112</td>
<td>95.6</td>
</tr>
</tbody>
</table>

Figure 6.5 Feature reduction rate comparison
6.6.2 Rejection and Acceptance Error Rate

Rejection of valid users or False Non-Match Rate is the likelihood that a valid user's keystroke is taken into account as a non-match for his reference template. Acceptance of invalid users or False Match Rate is the prospect that an individual's template is deemed a match for another entity's keystroke sample. The error rate can be arrived at from the following formula.

\[
\text{Error rate}(\%) = \frac{\text{Total no. of errors}}{\text{Total no. of samples}} \times 100
\]

Table 6.2 shows the values of the acceptance and rejection rates for both the wished-for model of authentication system and the existing system. The current system manifested for comparison is Keystroke Authentication using Bayesian classifier.

Table 6.2 Comparison of Error Rate

<table>
<thead>
<tr>
<th>Type of Error</th>
<th>Existing Method</th>
<th></th>
<th>Proposed Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Sample</td>
<td>Errors</td>
<td>Error (%)</td>
<td>Total Sample</td>
</tr>
<tr>
<td>Rejection of valid users</td>
<td>539</td>
<td>44</td>
<td>8.1</td>
<td>80</td>
</tr>
<tr>
<td>Acceptance of invalid users</td>
<td>768</td>
<td>22</td>
<td>2.8</td>
<td>80</td>
</tr>
</tbody>
</table>
From the foregoing Table 6.2 and Figures 6.6 and 6.7, it could be said that the proposed method is more professional as the percentage of error rate is very fewer.
6.7 GENETIC ALGORITHM

An algorithm was formulated to retrieve the Virtual Key Force through feature extraction before the pre-processed results are programmed to the next step, feature subset selection. From the reduced set of genes obtained in the previous pre-processing stage, a **wrapper approach** is used that combines a GA and an SVM to accomplish the feature subset selection. Feature subset selection is applied to high dimensional data before proceeding to the classification step. The basic idea consists in using a GA to discover “good” subsets of genes, the goodness of a subset being evaluated by an SVM classifier on a set of training data. All through this stage, prominent quality gene subsets are recorded to an archive for further breakdown. At the end of the GA, the analysis of the archived gene subsets is performed: gene subsets are weighed against one another and the predominantly and recurrently appearing genes are identified.

Feature subset selection is fundamentally an optimization intricacy, since this is connected with searching the space of possible features to recognize one that is optimum or near-optimal in accordance with some performance measures, since the objective is to acquire any subset that reduces or to improve a particular measure. GA wrapper is preferred over ensemble since the former generates a population of accurate classifiers. Genetic Ensemble Feature Selection (GEFS) adds a diversity term in the fitness function of GA.

Ensemble is a set of classifiers trained differently: by discrete data sets, by diverse features or by separate models. After individual classifiers are trained, they are combined by either majority voting or averaging to output a single value. The performance of an ensemble classifier has been found to be quite high in practice in a variety of applications. Individual classifiers
participating in an ensemble have to be accurate as well as diverse in order to result in accurate ensemble. It is but natural to combine.

6.8 PARTICLE SWARM OPTIMIZATION

An exclusive technique is proposed in chapter 5 based on the emotion of the user for authentication using the keystroke dynamics. The emotions of the user who is typing are extracted to identify the right user. In the projected methodology, the feature extraction process is employed to extract various emotional features from the user in the process of authenticating the approved user. Feature extraction is followed by feature selection process where the most pertinent features are selected. Particle swarm optimization (PSO) is used for the feature selection process which is an optimization algorithm that performs the selection of better particle i.e. features required for the planned approach.

PSO originated from the simulation of social behaviour of birds in a flock. In PSO, each particle flies in the search space with a velocity adjusted by its own flying memory and its companion’s flying experience. Each particle has its objective function value which is decided by a fitness function. PSO is an evolutionary computation technique which is very similar to that of the Genetic Algorithm where a particular system is initialized by a population of random solutions.

6.8.1 Keystroke Features

The foremost feature extracted is the keystroke features which is derived from the typing nature of the particular user. The keystroke features can be extracted from the process of typing any data by the users in the keyboard. Keystroke events were branched out as key down and key up events. The character representation of the key was extracted at run-time as
this proved to be a difficult task to perform offline. The current key state is also recorded for each new key event which is represented by a set of integers for each key on the keyboard. The keystroke features are apportioned into single key features and compound key features. The single key features are of divergent types like features that are summary features of the complete sample text and those features that are created for each individual key on the keyboard. The single key features were employed for extracting the keystroke features.

These features are extracted from the keystrokes of the users. The ‘D2D_AllKeys’ is weighed up as a major feature and this is the time duration of the user in pressing one key down to the next key down is extracted which provides the speed in which a person types in the keyboard precisely. Each pressing of a key can take different time duration depending on the pressing speed of the user and this can provide a distinct feature to establish the proper user. Based on this duration, the rest of the features can also be extracted. ‘KeyDur_ [KEY]’ is used to extract the duration between one key down and the next key up as well as the calculation of the time duration between one key down to up. U2U_AllKeys is the feature which measures the distance between one key up to next key up. ‘NumChars’ is the feature where the total number of the sample is estimated and the entire typing process time duration is extracted with the number of characters.

6.9 GENETIC ALGORITHM VS PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are the two evolutionary computational techniques and these are population-based search methods.
Both Genetic Algorithms and Particle Swarm Optimization share common elements.

i) Initialize a population in a similar manner.

ii) Use an evaluation function to determine how fit a potential solution (particle) is.

iii) They are generational, that is, both repeat the same set of processes for a predetermined time.

Here in PSO along with each potential solution, randomized velocity are also assigned which constitute a particle. Each particle follows its coordinates in the problem space in connection with the best solution. The fitness value is also considered for further processing. This fitness value is referred to as pbest. The location of these solutions is regarded as gbest. The PSO thus provides a better solution.

Several positive and convenient aspects merit the attention in the choice and successful employment of PSO.

- Particle Swarm Optimization is conveniently dissimilar to other genetic algorithms, evolutionary programming and evolutionary strategies in the absence of selection process
- All the particles in PSO are kept as members of the population through the course of the run
- PSO is the only algorithm that does not implement the survival of the fittest principle
- It is an additional feature that there is no cross over operation in PSO
6.9.1 Performance Analysis

The performance analysis of proposed PSO method of user authentication is done based on the feature reduction rate. The feature reduction rate of proposed PSO method is compared with that of the existing method (GA). The feature reduction rate is calculated with the help of the following expression shown.

\[ F_r = \frac{F_t - nF_s}{F_t} \]

In this expression, \( F_r \) represents the feature reduction rate, \( F_t \) represents the total number of features and \( nF_s \) represents the number of features selected. The feature reduction rate is calculated based on the number of features selected. The ratio of difference between the number of features used and the selected features to that of the total number of features gives the respective value. Based on these values, the feature reduction rate is calculated. The obtained value is compared with the existing method and the results show that the proposed method exhibits better feature reduction rate. The Table 6.3 shows the feature reduction rate value for proposed PSO method as well as the existing GA method. From the values, it is clear that proposed PSO method provides better reduction rate than that of the existing GA method (Figure 6.8).

Table 6.3 Comparison table for Feature reduction rate of Proposed PSO method with the existing GA method

<table>
<thead>
<tr>
<th>Methods</th>
<th>Feature Reduction rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing GA Method</td>
<td>95.6</td>
</tr>
<tr>
<td>Proposed PSO method</td>
<td>96.3</td>
</tr>
</tbody>
</table>
CONCLUSION

Keystroke features like dwell time, flight time, di-graph, tri-graph and virtual key force of every user are used in the keystroke dynamics. A GA based wrapper approach to be applied to keystroke dynamics based authentication is proposed. One class of SVM is used as base classifier and forced diversity through the uniqueness of each chromosome. This renders a rather complicated post processing unnecessary. By comparing the GA method to other existing methods, the feature reduction rate is very high and also the error rate is very small. User authentication based on keystroke dynamics is concerned with accepting or rejecting someone based on the way the person types. In PSO method, emotional status is employed as a biometric along with the keystroke dynamics. PSO method aims mainly the emotions that the user has while entering the text using the keyboard. The results of
PSO method show an improved authentication of the user than that of GA method in analogy. A thorough investigation and analysis also point to the fact that with regard to the difference in computational effect between PSO and GA is problem dependent. For instance, it is probable that PSO outperforms GA with a larger differential in computational efficiency when used to solve unconstrained non linear problems with continuous design variables and less efficient differential when applied to constrained non linear problems with continuous or discrete design variables. Three essential postulations should be borne in mind when selecting a model for each optimization problem:

- Choice of an appropriate objective function is mandatory
- Trial and error approach is essential
- No single metaheuristic process is to be used for solving a problem: hybrid approach would be the best option

The results were compared with my two publications. I have published two papers on Keystroke dynamics using GA and PSO by making an extensive study based on more than 30 technical papers published in various journals. Hence, the final results were compared with my own publications which reflected the improvement obtained by PSO over GA.