Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Hybrid Solution for the Feature Selection in Personal Identification Problems through Keystroke Dynamics

Gabriel. L. F. B. G. Azevedo, George D. C. Cavalcanti and E. C. B. Carvalho Filho

Abstract— **Techniques based on biometrics have been successfully applied to personal identification systems. One rather promising technique uses the keystroke dynamics of each user in order to recognize him/her. In this work, we present the development of a hybrid system based on support vector machines and stochastic optimization techniques. The main objective is the analysis of these optimization algorithms for feature selection. We evaluate two optimization techniques for this task: genetic algorithms (GA) and particle swarm optimization (PSO). In the present study, PSO outperformed GA with regard to classification error and processing time, but was inferior regarding the feature reduction rate.**

I. Introduction

Information security systems based on keystroke dynamics have been currently attracting considerable attention as a cheap solution that is more transparent to the user than fingerprint or iris recognition [1]. In this type of system, a user is identified by a login, a password and a keystroke pattern, the latter of which is a unique trait that is far more difficult to copy than the others and runs no risk of ever being lost. This system could be used in any setting where the user needs a password in order to gain access to particular information and/or functionalities, including Internet access [2].

Personal verification systems through some type of behavioral biometric characteristic, such as signature, voice or keystroke dynamic, are normally constructed from a set of examples that define the user. In the case of keystroke dynamic, it would be necessary for the user to type his/her name or password a number of times in order for the system to be able to extract the relevant features that uniquely represent the user. However, the task of typing one's name over and over is both tiring and tedious, which could lead users to alter their normal pattern. Thus, most systems based on biometrics are required to work with a summarized set of information from which to extract knowledge. In order to reduce this problem, we could eliminate some features of the original dataset, selecting only the best ones in terms of class cohesion. Another advantage that emerges is: reducing the number of features to be learned (thereby increasing the accuracy of the system), we also diminish the processing time required. One way to reduce this set of information would be to have a specialist delete the less representative data, but such a task is subjective and subject to errors [1]. It is expected that a system of this type should be executed automatically, where the user only has the task of typing his/her data repeatedly. The entire recognition process is accomplished by the system automatically. In the present work, we propose a feature selection approach based on Particle Swarm Optimization (PSO) [3], [4], [5], which is an evolutionary computation optimization technique based on the flocking behavior of birds. The results are compared with a technique based on genetic algorithms (GA) that was previously investigated by Yu and Cho [1].

In the next section, the Support Vector Machine (SVM) model is introduced and in the Section III, the PSO algorithm is introduced. In Section IV, it is presented an approach used to feature selection. Section V presents the methods used in the pre-processing of the data. Section VI describes the acquisition and division of the database. The experiments performed and their results are presented in Sections VII and VIII, respectively. Finally, in Section IX, we have the conclusions and suggestions for further studies.

II. Support Vector Machine

In recent works, such as [6] and [7], SVM models have been proposed to solve novelty detection problems. The basic idea of this approach is the creation of a region that encloses the vast majority of data pertaining to a particular class, making it possible that some authentic patterns of the class are not contained within this region. The model would generate the smallest possible region that could enclose the majority of data. Besides that, it would define a particular threshold that indicates how far a particular pattern could be outside this region and still be considered an authentic pattern. The idea behind SVM is to map the input dataset in a high-dimensional feature space, corresponding to the kernel function, and then separate them from the origin with a maximum margin. The algorithm would find a function f that returned $+1$ in the region that encompasses the majority of patterns contained in the input dataset, and -1 for data outside this region. After finding this function, any new pattern that was presented to this algorithm would have its value $f(x)$ determined by the indication of which side of the hyperplane it was found in the feature space (returning $+1$ for patterns considered authentic and -1 for novel patterns), as shown in Figure 1.

This work was supported in part by the Brazilian National Research Council CNPq (Proc. 478534/2006-0).

Authors are with Center of Informatics (CIn), Federal University of Pernambuco (UFPE), P.O. Box 7851, Cidade Universitária, Cep: 50.740-530 – Recife – PE – Brazil (corresponding authors to provide phone: 55-81-21268430;e-mails: glfbga@cin.ufpe.br; gdcc@cin.ufpe.br, ecdbcf@cin.ufpe.br).

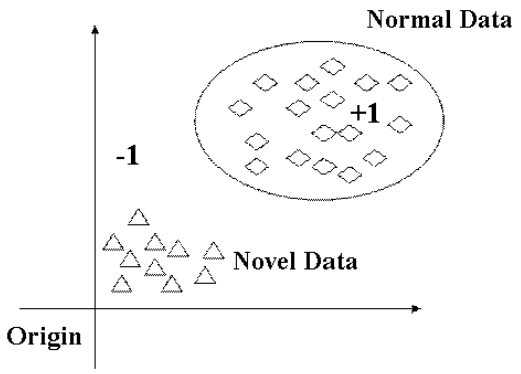

Figure 1 – SVM model for novelty detection.

The most used kernel function in novelty detection problems is Gaussian, given by Equation 1:

$$K(x, y) = \exp\left(-\|x - y\|\right)^2 / 2\sigma^2, \qquad (1)$$

where the parameter σ is responsible for the variation in the different estimates of regions in which the authentic data would be contained. It is important to point out that the Gaussian kernel ensures Mercer's condition, in which, for every value of σ, $K(x,y)$ corresponds to the internal product between patterns x and y, after these patterns have been mapped for the set of features, as we can see in Equation 2.

$$K(x, y) = (\Phi(x) \cdot \Phi(y)). \qquad (2)$$

The regions founded by the SVM algorithm are determined by the training dataset. These regions are determined by border patterns, if a pattern is inside the regions (authentic pattern), otherwise it is outside the regions (novelty). The data that compose this subset are called support vectors.

Without going into great detail, in order to separate authentic patterns from unauthentic ones, the SVM algorithm attempts to optimize the following expression:

$$\min_{\alpha} \sum \alpha_i \alpha_j k(x_i, x_j), \qquad (3)$$

subjected to the following restrictions:

$$\sum_i \alpha_i = 1, \quad \forall i, 0 \le \alpha_i \le \frac{1}{vd}, \qquad (4)$$

where d represents the dimensionality of the data and v is a free parameter that acts as both the upper bound for patterns considered fraudulent and the lower bound for the support vectors [8]. v can take on values between $1/n$ and 1, where n is the number of patterns. The α_i variables represent the Lagrange multipliers; there is one for each pattern contained in the original set. The task of the algorithm is to find the values of these coefficients. It is important to stress that only the Lagrange multipliers associated to the set of support vectors have values different from zero. When these multipliers are obtained, the decision function is defined as:

$$h(x) = \sum_{i \in SV} \alpha_i k(x_i, x) - \rho, \qquad (5)$$

where the sum is only valid for the multipliers associated to the support vectors. ρ is the threshold, which is calculated during training (for further details, see [8]).

III. Particle Swarm Optimization

PSO was introduced in the mid 1990s by Kennedy and Eberhart [4] and is a stochastic search technique that aims to optimize an objective function. It was developed through attempts to graphically simulate the movement patterns of birds in searching food. Later, in the search for theoretical foundations, studies were carried out on the manner in which individuals in society generally interact, exchanging information and reviewing their concepts in the search for better solutions to their problems [5].

In PSO, a population (swarm) of solutions (particles) is maintained. Let s be the size of the swarm, n be the dimension of the problem and t be the current instant, each particle $1 \le i \le s$ has a position $x_i(t) \in R^n$ in the space of solutions and a velocity $v_i(t) \in R^n$ that governs the direction and range of its movement. Moreover, each particle has a memory $y_i(t) \in R^n$ of the best individual position visited until instant t, and the swarm has a memory of the best position it has visited so far.

As the algorithm runs, the velocity of each particle is calculated according to two main references: the best individual position visited $y_i(t)$ (cognitive term of optimization) and the best position visited by the swarm $\hat{y}(t)$ (social term of optimization). After calculating the velocity by Equation 6, the position of the particle is updated by Equation 7.

$$\upsilon_i(t+1) = \omega \upsilon_i(t) + c_1 r_1 (y_i(t) - x_i(t)) + c_2 r_2 (\hat{y}(t) - x_i(t)), \qquad (6)$$

$$x_i(t+1) = x_i(t) + \upsilon_i(t+1). \qquad (7)$$

Item w is the weight of inertia (term of momentum), which multiplies the velocity of the previous instant t and makes the search more explorative at the beginning and more exploitive at the end, as its value generally drops linearly from 0.9 to 0.4. The terms $r1$ and $r2$ are random variables taken from U1(0, 1) and U2(0, 1), respectively. Both possess the function of randomizing the influences of each term of expression (individual and global). The coefficients of personal and global acceleration, $0 < c1, c2 \le 2$, respectively, generally possess fixed, equal values and are responsible for controlling how far a particle will move in a single iteration.

The best position visited $y_i(t)$ of each particle is updated by Equation 8, whereas the updating of the best position visited by the swarm is accomplished by Equation 9.

$$y_i(t+1) = \begin{cases} y_i(t), & se\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1), & se\ f(x_i(t+1)) < f(y_i(t)) \end{cases}, \quad (8)$$

$$\hat{y}(t+1) = \arg\min\ f(y_i(t+1)), \quad 1 \leq i \leq s. \quad (9)$$

The calculated velocity $v_i(t+1)$ in each cycle is generally limited to the interval $[-v_{max}, v_{max}]$, with $v_{max} = k.x_{max}$, $0.1 \leq k \leq 1.0$, which does not ensure that the particle is always within the boundaries of the search space, but limits the maximum distance the particle can move in a single iteration. Figure 2 displays the complete algorithm of the PSO pattern. The characteristics frequently attributed to PSO are rapid convergence into unimodal functions, with a good success rate, and the premature convergence into multimodal functions, with a high correlation between components.

```
Create and initialize a swarm of particles
repeat
  for each particle I of swarm do
    if f(x_i(t)) < f(y_i(t)) then
      y_i(t) = x_i(t);
    end if
    if f(y_i(t)) < f(ŷ(t)) then
      ŷ(t) = y_i(t);
    end if
  end for
  Perform PSO updates of which particle using
  equations (6) and (7)
until stopping criterion is met
```

Figure 2 – PSO algorithm

IV. FEATURE EXTRACTION

To capture keystroke dynamic, it would be necessary for users to type their own name a number of times. The system would set about capturing these features using two methods regarding the time (in milliseconds) that a particular user maintains the key pressed and the time elapsed between releasing one key and pressing the next (latency time). Hence, we have, for example, a pattern that represents the user "ABCD", illustrated in Figure 3, formed by a size 7 timing vector. The first element of the vector is the length of time in which the first key remains pressed; the second element is the latency time between releasing the first key and pressing the second; the third element is the length of time in which the second key remains pressed, and so on. In general, for a word of size n, a vector will be formed made

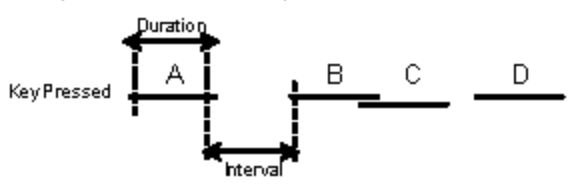

Figure 3 - Timing vector corresponding to a pattern "ABCD".

up of features the size of $2n-1$. In this case, we have n features that represent the lengths of time in which the keys remained pressed and $n-1$ features that represent the time elapsed between consecutive keys. In the present work, each pattern is formed by a vector with these lengths of time in milliseconds.

V. PRE-PROCESSING OF THE DATA

In the pre-processing step of the features, a clean-up of the data is performed in order to improve the performance of the system. In the present work, this step includes data normalization and feature selection. We use the sample standard deviation of the features to normalize the data [9]. It is important to point out that these methods are performed by class in order to create a system that is customized to the user.

As mentioned above, most pattern recognition systems based on behavioral biometrics (online and offline signatures, keystroke dynamic, etc.) have few data with which to represent the classes and a considerable amount of features to learn. The idea of the selection task is to obtain a subset of features that allows better rates of correct identification (with increased system performance) than with the entire set of features. This subset is obtained automatically. The user only has the work of presenting the training data to the system. The system then generates the feature selection model.

Feature selection is an optimization problem, since the aim is to obtain any subset that minimizes a particular measure (classification error, for instance). As we know, an optimization problem can be solved through stochastic algorithms. In [1], the authors used genetic algorithms (GA) to perform the selection task. In the present study, we propose an approach based on particle swarm optimization (PSO) algorithm.

In [8], it was shown that the feature selection task can be classified into two categories: filter and wrapper. The filter approach does not depend upon the algorithm used for verification, whereas the wrapper approach does depend upon this algorithm. Therefore, the filter approach is far more efficient (in terms of computational time) than the wrapper approach, as there is no need to train a particular algorithm numerous times with various possible subsets of features. Selection through a filter simply discards irrelevant features through a given measure. In [9], for instance, the features that presented the largest coefficients of variance ($CV = \sigma / \overline{x}$) were discarded because they exhibited considerable variation between patterns, which tends to hinder the recognition task. In the wrapper approach, the idea is to generate several subsets of features, evaluate them through a given algorithm and choose the subset that represents the best solution. This approach achieves better results than the filter approach, but is only viable if the algorithm used to train the classifier is efficient.

In the present work, we present a feature selection model that uses PSO as the optimization algorithm and SVM as the verification algorithm due to its known accuracy in pattern recognition problems [10]. The proposed model follows the structure displayed in Figure 4

In [1], GA was used to optimize a vector with binary features, where 1 indicates the presence of the feature and 0 indicates its absence. Each particle of a PSO is formed by a vector of real numbers and the algorithm optimizes this vector. In the present work, therefore, we decided to use this vector of real numbers as the likelihood of a feature being present in the subset, because these numbers represent the influence about the features. Thus, we normalize their values between 0 and 1. We also define a likelihood threshold that indicates that the feature is present in the subset if it has a value equal to or greater than this threshold and is absent otherwise. Thus, the final vector is formed by binary characters, as presented in [1]. In the present work, the function to be minimized is represented by the general classification error, which is calculated from the sum of the FAR (False Acceptance Rate) and FRR (False Rejection Rate) obtained by the SVM for the subset of features.

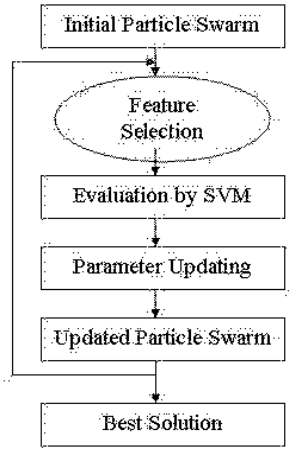

Figure 4– Hybrid model for feature selection through SVM and PSO.

VI. Database Acquisition and Division

The database used in the present work was the same used in [9]. The data were obtained through a software program developed in the Visual Basic language, which retrieves the necessary information that defines the keystroke dynamic of each user. The database used in this project is made up of information from 24 different users. Each user entered his/her complete name approximately 60 times. To obtain the data that represent frauds against the system, some users entered the complete names of other users approximately 5 times. The database generated here has approximately 60 authentic patterns and 40 false patterns for each of the classes.

In the present work, each user entered his/her complete name, which resulted in large vectors. For some classes, the input vectors had sizes greater than 70. The database used here was divided into 40 training patterns, 10 validation patterns and the remaining patterns were used for testing. As the database had a few classes containing less than 60 patterns (some with as little as 56 patterns, resulting in just 6 test patterns, which would have been of little significance), we opted to discard classes with less than 59 patterns in the data set. Thus, four classes were discarded, resulting in a total of 20 classes used.

VII. Experiments

The experiments conducted in the present work attempted to illustrate that PSO can be used as an alternative to GA in feature selection problems. As in [1], we used a basic GA and tested two possible PSO configurations. The verifier was the SVM as proposed in [6], [7]. After specifying the parameters of the verification and optimization algorithms, we performed 30 executions of the experiment, which included the presentation of the training data, feature extraction, pre-processing of the data (normalization and feature selection) and the verification step (through the test set). The system evaluation proposed was performed using three measures: classification error, processing time in seconds and the reduction percentage of the set of features.

The classification error is presented by two measures: FAR, when the system accepts an imposter (or fraud) as authentic, and FRR, when the system rejects an authentic user.

Another evaluation measure used in the present work regards the feature reduction capacity of the entry set. For example, if we have a pattern that initially has 60 features and after concluding the selection step, this pattern has 36 features, we have achieved a reduction of 40%.

As mentioned above, the verifier used in the present work was based on support vector machines. The kernel function used is Gaussian function, the most commonly employed for pattern recognition work. The value of the parameter σ used in this equation was obtained empirically and is 0.015. The value of v was also obtained empirically and is also 0.015.

The only parameter from the genetic algorithm that varied was the population size, which reached values of 20, 30 and 40 individuals. Each individual of the population was represented by binary vector. The crossover rate was 0.8 and the function used was the two-point crossover. The mutation rate was 0.1 and its function was uniform.

For the PSO algorithm, we used two configurations: one that considers the personal global acceleration coefficients, C_1 e C_2, to be 1.5 and another that considers them to be 2.0. These values were obtained empirically. For the inertia weight w (momentum term), the values drop linearly from 0.9 to 0.4 [3]. As with GAs, the PSO algorithm can be executed a maximum of 1000 iterations. The threshold values, which indicate the minimum likelihood that a feature must possess in order to make up part of the feature subset, were 0.7, 0.8 and 0.9.

VIII. RESULTS

In this section, we present the results obtained from our experiments. The Table 1 displays the results from the SVM verifier using a GA as the evolutionary algorithm for the feature selection task. The Tables 1 and 2 display the results achieved from PSO combined with SVM. From Table 1, we can see that the best results presented (regarding classification error, execution time and reduction rate) from the genetic algorithms were for the population composed of 40 individuals. In this case, the total error was 5.19%, having presented a FAR of 0.43%. The average execution time was 3.6s and the reduction rate was 52.49%. We can see that GAs present better results as population size increases. This is particularly true with regard to execution time, where the average exhibited a reduction of over 50%, dropping from 7.67s to 3.6s with the increase in population from 20 to 40 individuals. However, we can also see that the reduction rate of the feature set remains virtually unaltered throughout the experiments at approximately 52%.

Table 1 - Results obtained from the GA pattern.

Pop	FRR(%)		FAR(%)		class.error (%)		Time (s)		Reduction(%)	
	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev
20	7.11	6.41	0.53	0.95	7.64	6.63	7.67	12.95	52.29	2.72
30	5.65	5.87	0.40	0.70	6.05	6.02	4.75	8.49	52.16	1.88
40	4.75	4.89	0.43	0.82	5.18	5.11	3.60	7.27	52.49	2.36

Table 2 displays the results obtained for PSO with personal and global acceleration at 1.5. The best result for this algorithm was obtained from a PSO with a threshold of 0.9 and a population containing 30 particles. The result here was better than the one achieved with the GA regarding classification error and execution time, but inferior with regard to feature reduction rate. The best configuration achieved a 2.21% classification error, with a false positive error of 0.14%. Average execution time was 0.66s, which was the worst time result of these experiments, through still far below that achieved by the GA. The reduction rate was 46.67%, the second best rate obtained here. It should be pointed out that there was a considerable improvement with regard to feature reduction rate as the threshold rose, going from approximately 27.5% to approximately 46.5%.

Table 2 - Results obtained with PSO 1.5.

Thr-Pop	FRR(%)		FAR(%)		class.error (%)		Time (s)		Reduction(%)	
	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev
0.7-20	7.33	8.07	0.05	0.21	7.38	8.09	0.34	0.31	27.58	3.00
0.7-30	7.21	9.67	0.10	0.32	7.31	9.65	0.42	0.22	27.69	2.10
0.7-40	5.74	6.98	0.03	0.13	5.77	6.96	0.51	0.14	27.28	1.33
0.8-20	3.21	4.04	0.35	0.91	3.56	4.31	0.26	0.20	36.60	2.70
0.8-30	3.42	5.43	0.12	0.50	3.54	5.50	0.39	0.41	36.30	2.55
0.8-40	3.80	4.67	0.22	0.46	4.02	4.77	0.49	0.50	36.14	2.10
0.9-20	2.24	3.11	0.18	0.48	2.42	3.37	0.52	1.21	47.30	2.93
0.9-30	2.07	3.21	0.14	0.50	2.21	3.35	0.66	1.72	46.67	2.56
0.9-40	2.45	3.80	0.33	0.53	2.78	3.93	0.60	1.21	46.50	2.00

Table 3 displays the results obtained for PSO with personal and global acceleration at 2.0. The best result by this algorithm was obtained from a PSO with a threshold of 0.9 and a population containing 30 particles. The results achieved here were better than those obtained from the GA and the PSO with accelerations of 1.5 regarding classification error, but were inferior to the previous PSO regarding the execution time and inferior to the GA regarding feature reduction rate. The best configuration achieved a 1.58% classification error, with a 0.41% false positive error, which is higher than that of the previous PSO. Average execution time was 0.69s, the second worst time for these experiments and worst than that achieved by the previous PSO, but still better than that achieved by the GA. The reduction rate was 47.51%, which was inferior to the rates achieved by the GA, but better than those obtained by the previous PSO. An in the previous experiment, we see an improvement in the feature reduction rate with an increase in population.

Table 3 - Results from PSO 2.0.

Thr-Pop	FRR(%)		FAR(%)		class.error (%)		Time (s)		Reduction(%)	
	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev	mean	s.dev
0.7-20	5.09	5.62	0.08	0.28	5.17	5.67	0.32	0.26	31.22	2.94
0.7-30	4.80	6.15	0.02	0.07	4.82	6.14	0.42	0.26	30.06	2.47
0.7-40	6.14	7.15	0.06	0.26	6.20	7.10	0.51	0.16	30.12	1.82
0.8-20	4.42	5.28	0.03	0.08	4.45	5.28	0.27	0.27	38.89	3.65
0.8-30	3.41	5.01	0.03	0.10	3.44	5.04	0.36	0.40	37.46	1.89
0.8-40	3.51	4.85	0.29	0.83	3.80	5.24	0.45	0.39	37.76	1.86
0.9-20	2.06	3.20	0.41	0.88	2.47	3.56	0.60	1.70	47.34	2.69
0.9-30	1.18	1.43	0.41	1.14	1.59	2.18	0.69	1.75	47.51	2.60
0.9-40	2.92	4.66	0.07	0.20	2.99	4.67	0.74	1.82	46.68	2.49

The following three figures present graphs that relate the results obtained from the best configurations of the GA and PSOs for the three performance measures. Figure 5 shows a comparative graph of the classification errors in which we can clearly see that the results obtained by the PSO with accelerations of 2.0 were better than the other procedures in practically all classes. We can also see that the GA was either inferior or at least equal (in few situations) for all classes.

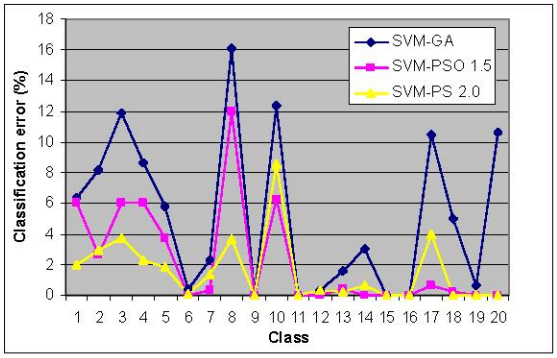

Figure 5– Graph of classification errors obtained by GA, PSO 1.5 and PSO 2.0.

Figure 6 displays the execution times obtained for each of the classes. We can see that the time obtained with the GA remains higher than that obtained by the two PSO configurations used here. We can also see that the times obtained by the two PSO configurations tested here are practically equal.

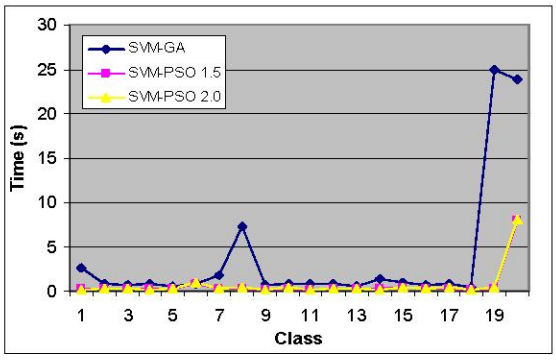

Figure 6 – Graph of processing times obtained by the GA, PSO 1.5 and PSO 2.0.

The Figure 7 displays the feature reduction rates obtained by the GA and PSO algorithms. Differently from the other two measures, the GA exhibited the best performance in this case, with a higher reduction rate that obtained by PSO in practically all classes.

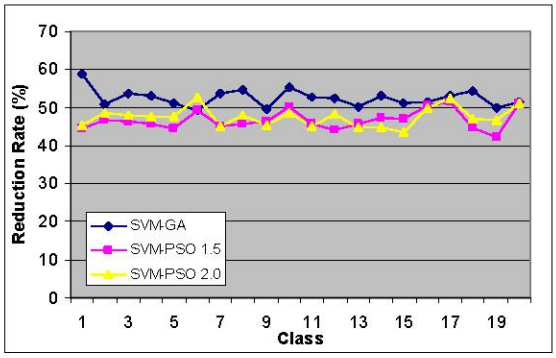

Figure 7 – Graph of feature reduction rates obtained by the GA, PSO 1.5 and PSO 2.0.

IX. Conclusion

As we can see in the previous section, the PSO evolutionary algorithm showed a suitable solution to the feature selection task. In the experiments carried out in the present work, PSO was superior to the GA pattern in two of the three performance measures evaluated. It was superior to the GA regarding classification error and processing time, while it was inferior regarding feature reduction rate. Between the two PSO configurations tested, that which was defined with personal and global acceleration at 2.0 obtained better results than that which was defined with personal and global acceleration at 1.5 regarding classification error and feature reduction rate, while it was inferior with regard to processing time. We can also see that, despite the former having obtained a better result regarding classification error, the latter obtained a lower FAR. It should be pointed out that in most identification systems, it is more important to have a lower FAR than a lower FRR.

In future work, we intend to explore further the use of PSO as an algorithm for feature selection, with the aim of further studying the variation of its parameters and their influence on the results. We also intend to test some techniques based on statistics, as in [9], to be executed before PSO. Another possible alternative is to dispense using thresholds in PSO and use the final likelihood directly on a particular feature. Thus, if a given feature possesses a likelihood of 0.8, its final value would be 80% of the original value. Yet another alternative would be to use auto-associative neural networks as verifiers, as, despite [2] having determined that these networks in conjunction with SVM are quite slow for the feature selection task, we see that PSO exhibited a processing time far below that of the GA.

References

[1] E. Yu and S. Cho. Keystroke dynamics identity verification - its problems and practical solutions. Computers and Security, 23(5):428–440, 2004.

[2] S. Cho, C. Han and H. Kim. Web-based keystroke dynamics identity verification using neural network. J Organ Comput Electron Commerce, 4(10):295–307, 2000.

[3] F. V. D. Bergh and A. P. Engelbrecht. A Cooperative Approach to Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation, Vol.8, No.3, 225-239, June, 2004.

[4] J. Kennedy and R. Eberhart. Particle Swarm Optimization, in: Proc. IEEE Intl. Conf. on Neural Networks (Perth, Australia), IEEE Service Center,Piscataway, NJ, IV:1942-1948, 1995.

[5] J. Kennedy and R. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 2001.

[6] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13, 1443-1471, 2001.

[7] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor and J. Platt. Support vector method for novelty detection. In Advances in Neural Information Processing Systems 12, 582–588, MIT Press, Cambridge, MA, 2000.

[8] J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. In: Motoda H, Liu H, editors. Feature extraction, in construction, and subset selection: a data mining perspective. New York: Kluwer; 44-49, 1998.

[9] G. D. C. Cavalcanti, E. Pinheiro, E. C. B. Carvalho Filho. A System of Verification of Personal Identity through Keystroke Dynamics (in portuguese). In Proceeding of the V Brazilian National Artificial Intelligence Meeting, pages 791–800, 2005.

[10] H. Byun and S. Lee. Applications of support vector machines for pattern recognition: a survey. Lecture Notes In Computer Science, Vol. 2388, 213-236, 2002.