Computers & Security (2004) **23**, 428–440

Computers & Security

www.elsevier.com/locate/cose

Keystroke dynamics identity verification—its problems and practical solutions

Enzhe Yu*, **Sungzoon Cho**

Department of Industrial Engineering, College of Engineering, Seoul National University, San 56-1, Shillim Dong, Kwanak-Gu, Seoul 151-744, Republic of Korea

Received 5 November 2003; revised 2 February 2004; accepted 10 February 2004

KEYWORDS
Keystroke dynamics identify verification;
Novelty detection;
Autoassociative MLP;
Support vector machine;
Feature subset selection;
Ensemble based on feature selection

Abstract Password is the most widely used identity verification method in computer security domain. However, because of its simplicity, it is vulnerable to imposter attacks. Use of keystroke dynamics can result in a more secure verification system. Recently, Cho et al. (J Organ Comput Electron Commerce 10 (2000) 295) proposed autoassociative neural network approach, which used only the user's typing patterns, yet reporting a low error rate: 1.0% false rejection rate (FRR) and 0% false acceptance rate (FAR). However, the previous research had some limitations: (1) it took too long to train the model; (2) data were preprocessed subjectively by a human; and (3) a large data set was required. In this article, we propose the corresponding solutions for these limitations with an SVM novelty detector, GA–SVM wrapper feature subset selection, and an ensemble creation based on feature selection, respectively. Experimental results show that the proposed methods are promising, and that the keystroke dynamics is a viable and practical way to add more security to identity verification.

Introduction

In typing a phrase or a string of characters, the typing dynamics or timing pattern can be measured and used for identity verification. More specifically, a timing vector consists of the keystroke duration times interleaved with the keystroke interval times at the accuracy of milliseconds (ms). If a password of n characters is typed, a $(2n+1)$ dimensional timing vector results, which consists of n keystroke duration times, $(n-1)$ keystroke interval times, and the return key (in most cases, return key is meaningless, thus ignored). Fig. 1 illustrates the timing vector when a string "*ABCD*" ($n = 4$) is typed. An actual example of a 9-dimensional timing vector is $[30, 60, 70, -35, 60, 35, 75, 40, 55]$. The time unit is in milliseconds. When a key is stroked before the previous key is released, the keystroke interval time is represented as negative (<0).

* Corresponding author.
E-mail addresses: enzhe@snu.ac.kr (E. Yu), zoon@snu.ac.kr (S. Cho).

0167-4048/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2004.02.004

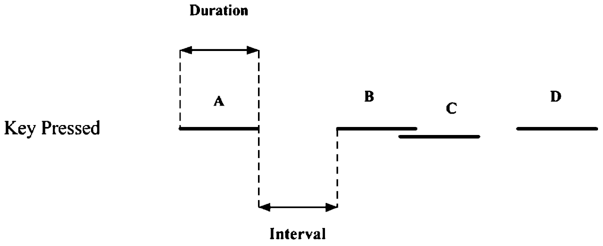


Figure 1 Timing vector corresponding to a string "*ABCD*".

Motivated by the observation that a user's keystroke pattern is highly repeatable and distinct from that of other users (Gaines et al., 1980), the owner's timing vectors were collected and used to build a model that discriminates the owner from imposters. If a future timing vector is found to be close enough to the model, it is classified as "accept" or belonging to the owner. If not, it is classified as "reject" or belonging to the imposter. Since the keystroke dynamics approach is purely software-based, it is less expensive and more user transparent than the currently popular methods such as the fingerprint-based and the iris-based authentication approaches. The keystroke dynamics approach can be even used over the Internet (Cho et al., 2000).

Keystroke dynamics identity verification can be applied to any circumstances where password-based access controls take place. For instance, a model may be embedded into a Window NT or Window 2000 log-in module as in http://www.biopassword.com, or it may be residing over the Internet as in Cho et al. (2000).

In practice, the authentication method based on keystroke dynamics can also be used to assure the user's ongoing presence at an input device. For instance, a user could be periodically prompted to type in his or her password to assure the presence.

Like other biometrics-based approaches, the keystroke dynamics identity verification has 2 types of error, i.e., false acceptance rate (FAR) and false rejection rate (FRR). FAR denotes the rate that an imposter is allowed access, and FRR denotes the rate that a legitimate user is denied access. Because 1 type of error can be reduced at the expense of the other, an appropriate trade-off point is commonly used as a threshold based on the relative cost of the errors.

In 1980, Gaines et al. first proposed the approach using keystroke dynamics for user identity verification. Experiments with a population of 7 candidates were conducted. Later on, Leggett et al. (1991) conducted similar experiments by applying a long string of 537 characters, reporting a result of 5.0% FAR and 5.5% FRR. Recently, through the use of neural networks, a comparable performance of 12%–21% was achieved using short strings such as real-life names (Brown and Rogers, 1993). Obaidat and Sadoun (1997) reported a 0% error rate in user verification using 7-character-long log-in name. In their experiments, however, both the imposter's typing patterns and the owner's patterns were used for training. This is unacceptable in practice since imposter's patterns can be obtained only when the password is revealed. Another limitation is that the training data set was too large to be practical (6300 from owner and 112 from imposters). Also, the training and test patterns were not chronologically separated. In Cho et al. (2000), a neural network novelty detection model was built by training the owner's patterns only, and the model was used to detect imposters using some sort of similarity measure, reporting a 1.0% FRR and 0% FAR. However, there were 3 main limitations in that research. First, the computational cost was too high. It usually takes hundreds, even thousands of seconds to build a novelty detector using a neural network. Second, subjective preprocessing was involved. In practice, raw user patterns contain much noise and many outliers because of the user's typing inconsistencies, which accordingly could result in poor detection accuracy. The previous research employed manual preprocessing to clean the data, and this is not acceptable in a real-world application. Third, a large number of training patterns were required, especially for a high performance neural network model. Hundreds of training patterns were used in Cho et al. (2000), but in practice, most users would be unwilling to type the passwords hundreds of times for data collection. One has to be able to build a good model with as few data as possible: fewer than 50, for instance.

In this paper, we propose a combination of approaches and models that could solve or alleviate the limitations mentioned above. First, support vector machine (SVM) (Schölkopf et al., 1999, 2000) is proposed for novelty detection, which performs as well as the neural network but requires much less computational cost, i.e. SVM only needs about 1/1000 of the neural network's training time. Second, a wrapper feature selection approach is employed (Yu and Cho, 2003), which can automatically select a relevant subset of features and ignores the rest, thus producing a better accuracy. In particular, genetic algorithm (GA)-based wrapper approach is employed. Third, an ensemble model based on feature selection

(FS-Ensemble) is proposed to alleviate the deficiency of a small training data set.

This paper is structured as follows. In the next section, the SVM model is introduced. Sections "GA—SVM wrapper approach for feature selection" and "Ensemble creation based on feature selection" lay out some limitations of the previous research, and propose the wrapper approach and the ensemble approach, respectively. Experimental settings and the results with regard to the proposed approaches are described in section "Experimental results". Finally, section "Conclusion" summarizes the result of this research and discusses the limitation and future research topics.

Identity verification using SVM novelty detector

User identity verification is a challenging task from a pattern classification viewpoint. It is a 2-class (owner vs. imposters) problem, but only the patterns from the owner are available in advance. Most previous researches (Brown and Rogers, 1993; Goldberg, 1989; Leggett et al., 1991; Obaidat and Sadoun, 1997) used both the owner's and imposter's patterns to train their models. Yet, this is not practical in real-world applications because there are millions of potential imposters, thus it is not possible to obtain all the prospective imposter patterns. Nor is it desirable to publicize one's password to collect potential imposter's timing vectors at the risk of fatal intrusion. The only solution is to build a model of the owner's keystroke dynamics and use it to detect imposters who are using some sort of similarity measures. This type of problem is known as a *partially exposed environment* or *novelty detection* (Dasarathy, 1979).

Previous research by Cho et al. (2000) has reported on the performance of neural network novelty detector superior to other methods, such as *k*-NN (nearest neighbors). Their research adopts a 2-layer AaMLP. Hwang and Cho (1999) found that the 2-layer AaMLP as a novelty detector is weak in modeling the data with nonlinear or multi-modal distributions. They proposed a 4-layer AaMLP to overcome such limitations. Later on, Yu and Cho (2003) applied the proposed model to keystroke dynamics-based identity verification problem, reporting on improved results. Although the proposed neural network seemed to be the best model thus far, it required too long a training time, i.e., 4—5 times more than that of 2-layer AaMLP. Thus, a model with both excellent performance and quick learning speed is desired.

Recently, support vector machine (SVM) gained great interests in pattern recognition. One reason is its theoretical foundation of structural risk minimization principle minimizing time consuming trial-and-error search of hyper-parameters, which is ubiquitous in the neural network model (Vapnik, 1995). Another reason for its success is its excellent classification performance in numerous application areas (Byun and Lee, 2002). Recently, Schölkopf et al. (1999, 2000) extended the support vector machine methodology to "1-class" classification, i.e., novelty detection problem. Their idea is to map the data into the feature space corresponding to the kernel, and then to separate them from the origin with a maximum margin. The algorithm returns a decision function f that takes the value $+1$ in a "small" region capturing most of the normal data, and -1 elsewhere. For a new point x, the value $f(x)$ is determined by evaluating which side of the hyperplane it falls on, in feature space (see Fig. 2).

Let $x_1, x_2, \ldots, x_l \in X$, where l is the number of normal data, X denotes 1 class and is a compact subset of R^N, and let Φ be a feature map $X \rightarrow F$, which transforms the training data to a dot product space F such that the dot product in the image of Φ can be computed by evaluating some simple kernel

$$k(x, y) = (\Phi(x) \cdot \Phi(y)). \quad (1)$$

A popular example of the kernel is the *Gaussian* kernel, which is

$$k(x, y) = e^{-\|x-y\|^2/s}. \quad (2)$$

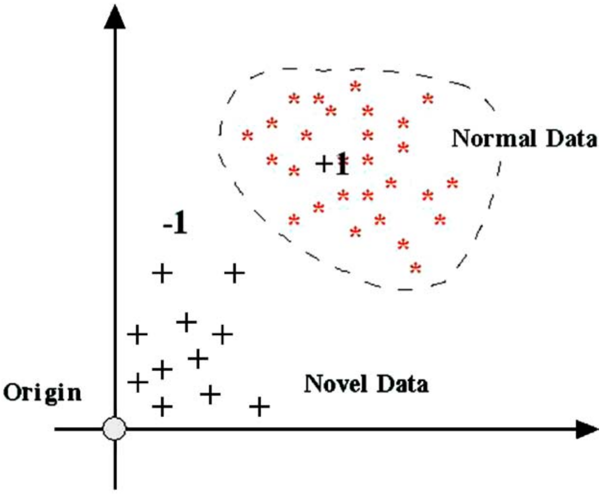


Figure 2 The 1-class SVM for novelty detection.

To separate the normal data from the origin, one needs to solve the following quadratic programming problem:

$$\min_{w \in F, \xi \in R^l, \rho \in R} \frac{1}{2}\|w\|^2 + \frac{1}{\nu l}\sum_i \xi_i - \rho \tag{3}$$

$$\text{subject to } (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0. \tag{4}$$

Since nonzero slack variables ξ_i are penalized in the objective function, we can expect that if w and ρ solve this problem, then the decision function

$$f(x) = \text{sgn}((w \cdot \Phi(x)) - \rho) \tag{5}$$

will be positive for most examples x_i contained in the training set, while the SV type regularization term $\|w\|$ will still be small. The trade-off between these 2 goals is controlled by $\nu \in (0, 1)$.

Deriving the dual problem, and using Eq. (1), the solution can be shown to have an SV expansion

$$f(x) = \left(\sum_i \alpha_i k(x_i, x) - \rho\right) \tag{6}$$

(patterns x_i with nonzero α_i are called support vectors), where the coefficients are found as the solution of the dual problem:

$$\min_\alpha \frac{1}{2}\sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \tag{7}$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1.$$

If ν approaches 0, the upper boundaries on the Lagrange multipliers tend to infinity, thus the problem then resembles the corresponding hard margin algorithm. If ν approaches 1, then the constraints only allow 1 solution, where all α_i are at the upper bound $1/(\nu l)$. In this case, for kernels with integral 1, such as normalized *Gaussian* kernels, the decision function corresponds to a *Parzen* windows estimator with a threshold.

GA—SVM wrapper approach for feature selection

Novelty detection models are built under the assumption that the owner's typing follows a consistent pattern. But there are always inconsistent typing patterns attributed to human error. One popular method of tackling the problem is manual preprocessing of data, i.e., either removing the variable(s) whose values are dispersed, or removing data samples which seem to be inconsistent with other patterns (Cho et al., 2000). However, practically, it is very difficult to correctly identify noisy data or outliers. Moreover, human judgment on inconsistency is quite subjective, and the criteria may vary from person to person. On the other hand, we cannot skip the preprocessing procedure because without data cleaning, according to observations on the experiments, the model performance would decline. Even the best-so-far model produced an error rate as high as 15% FRR when FAR was set to 0%.

Ideally, identity verification should have no human intervention to deal with the owner's raw patterns, and an automated process is desired. One way to improve the performance is to employ an "automated" preprocessing process, i.e., feature subset selection method. We call this an "automated" process, because once the feature selection module is built, all we need to do is simply input original data to the module. Then the selected feature subset will come out from the module automatically. More specifically, in an identity verification system, the identity verification model can be built automatically whenever the user's training data are presented. The conceptual frame is illustrated in Fig. 3.

Feature subset selection is essentially an optimization problem, which involves searching the space of possible features to identify a set of features that are optimum or near-optimal with respect to certain performance measures (e.g., accuracy, learning time, etc.). According to the characteristics of the search strategy, feature subset selection algorithms can be broadly classified into 3 categories (Yang and Honavar, 1998): (a) exhaustive search, (b) heuristic search, and (c) randomized search. Prominent among the randomized search algorithms is the genetic algorithm (GA), which does not require computation cost like the exhaustive search (a) and the restrictive monotonicity like the randomized search (b).

On the basis of the dependence of the induction algorithm, the feature selection procedures can be classified into 2 categories: *filter* and *wrapper* (Yang and Honavar, 1998). The filter approach somehow filters out "irrelevant" features and passes only "relevant" features to classifiers. It is performed independently of the learning algorithm, and in general, it is computationally more

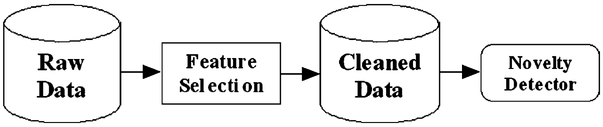


Figure 3 Outline of automated model building.

efficient than the wrapper approach. Its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm that is used to construct the classifier. The wrapper approach, on the other hand, tries many different sets of features by building a model using a learning algorithm, and then chooses the one with the best performance measure. Thus, it involves the computational overhead of evaluating numerous candidate feature subsets. This approach leads to a better result than the filter approach, but it is feasible only if the learning algorithm used to train the classifier is computationally efficient (Yang and Honavar, 1998).

In this paper, we employed a GA paradigm for the randomized search and SVM as a base learner in the wrapper approach. A population of feature subsets would evolve through the mechanism of the GA, and a feature subset would be evaluated through training and testing an SVM with the data set. GAs are stochastic search techniques based on the mechanism of natural selection and genetics, and they are generally quite effective for rapid global search of large search spaces in difficult optimization problems (Goldberg, 1989). Previous researches have reported on the feasibility of GA for wrapper approach to feature subset selection (Yang and Honavar, 1998). SVM also suits well as a base learner because of its fast training capability. An initial population is made up of diversified binary strings indicating the features selected. These candidates undergo crossover and mutation, evaluated by the SVM-based learner. Only those that are selected according to the specified multi-criteria fitness are put back into the population, and the process is repeated for a fixed number of generations. The best solutions are obtained in the end (see Fig. 4).

In the proposed GA—SVM wrapper approach, a *Gaussian* kernel was used for SVM, and the parameters of SVM were tuned through a heuristic method. The GA was implemented with the following settings. The chromosome was a binary string where each bit denotes whether the corresponding feature is *present* (1) or *absent* (0). The population size was generally set at 30, however, when the population diversity resulted in an unsatisfactory performance, it was modified up to 50. The crossover rate of 0.6 and the mutation rate of 0.01—0.02 were adopted with the corresponding mechanisms being 2-point crossover and uniform mutation, respectively. Selection provides the driving force in the evolutionary process, and the selection pressure is critical. If the pressure is too high, the search will terminate prematurely; on the other hand, if the pressure is too low, the searching progress will be slower than necessary.

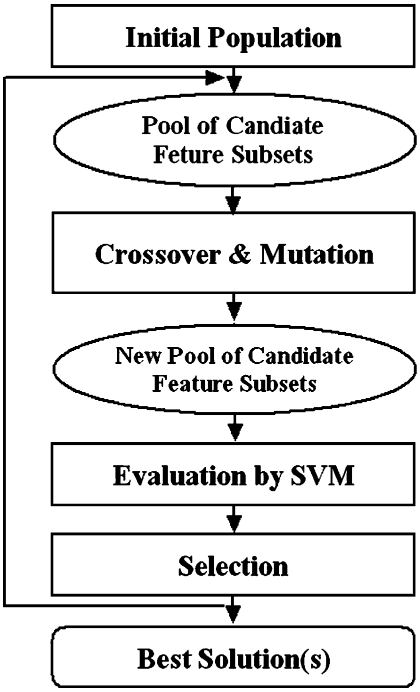


Figure 4 GA—SVM wrapper feature subset selection.

At the early stage of evolution, a low selection pressure is preferred for a wide exploration of the search space. At the end of evolution, however, where the population is near convergence, a high selection pressure is adopted to exploit the most promising regions of the search space. As for the sampling space, we adopted an enlarged sampling space, where both parents and offspring have the same chance of being selected because genetic operations are blind in nature and the offspring may not always be better than their parents. With a fixed population size of *M*, *M* parents and *N* offspring will be involved in the sampling space with its size ($M+N$). These parents and offspring will compete for survival, and *M* best members will be selected and used as parents of the next generation. The sampling mechanism followed the probabilistic roulette wheel selection. To discriminate among the similar strong individuals in the last 10%—20% generations, a linear scaling method was applied to deal with the selection probability.

The fitness function combined 3 different criteria, i.e., the accuracy of the novelty detector, the learning time, and the dimension reduction ratio, and the fitness function takes the following form:

$$\mathrm{Fitness}(\boldsymbol{x}) = \alpha \mathrm{Acc}(\boldsymbol{x}) + \beta \frac{1}{\mathrm{LrnT}(\boldsymbol{x})} + \gamma \frac{1}{\mathrm{DimRat}(\boldsymbol{x})}, \quad (8)$$

where Fitness(x) is the fitness of the feature subset represented by x, Acc(x) is the test accuracy of the SVM novelty detector using the feature subset represented by x, LrnT(x) is the time taken to train the SVM, and DimRat(x) is the dimensionality reduction rate. If the dimensionality of full feature set was 15, and the dimensionality of currently selected feature subset is 6, for instance, then DimRat(x) $= 6/15 = 40\%$. Given similar accuracies, we would choose the feature subset with smaller DimRat(x), i.e., more compact feature subset. The coefficients α, β, and γ were arbitrarily selected according to the importance of the terms, i.e., Acc(x), LrnT(x) and DimRat(x). In our case, we placed more emphasis on model performance, and focused less on learning time and dimension reduction rate. The values of α, β, and γ were set as 10, $\frac{1}{100}$, and 1, respectively.

Ensemble creation based on feature selection

In actual application, the keystroke dynamics identity verification system often starts with an insufficient amount of data since users are unwilling to type the passwords hundreds of times for data collection, let alone thousands of times. Thus, one should build a system with much fewer than 100 patterns in the beginning, and then gradually increase the number of patterns. However, insufficient data could lead the identity verification model to a low accuracy.

A classic learning curve achieved from previous empirical studies is presented in Fig. 5, showing that model accuracy increases strongly with the number of training data until an adequate number of training data is observed and then the improvement tails off (Smyth and Cunningham, 1996). A reasonable conjecture would be that overfitting might be a problem to the left of the knee point in the graph. With the complexity of the problem increasing, this knee point may move more toward right. When the number of the owner's training patterns is limited, overfitting will always occur. For experimental observation, we used an AaMLP to conduct experiments on a password ID, which consists of about 200 samples. Experimental results showed that with 5% of training data, the accuracy was only around 35.38%. Between the starting point and the knee point, i.e., 5%—45%, the accuracy increased very quickly. After this point, the accuracy increased slowly and finally converged around 98.41%. This experimental result is shown in Fig. 6.

With a limited number of training data, the models mostly tend to overfit, and thus result in a poor performance. One way to alleviate such a problem is to employ the ensemble method (Breiman, 1996; Shapire, 1990; Yao and Liu, 1998). There are basically 2 requirements on ensemble creation: (1) diversity, i.e., the classifiers should be as diverse as possible; and (2) accuracy, i.e., the candidate classifiers should more or less perform well. Recently, Ho (1998) presented a method for ensemble creation based on random feature selection to solve the overfitting problems which originated from insufficient data. One obvious limitation of this method is that it emphasized diversity too much while it ignored accuracy. Thus the model performances tend to be mixed. A balance between the degree of diversity and accuracy is desired for ensemble creation.

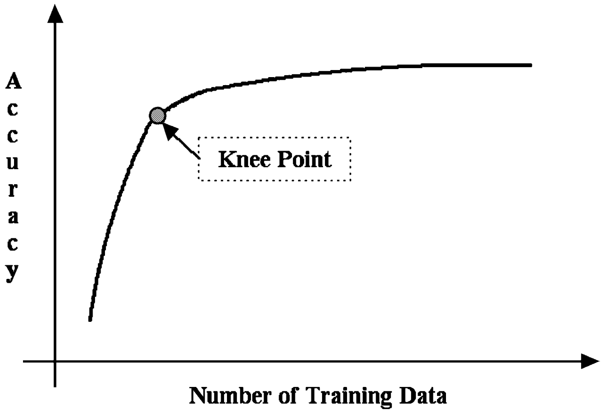


Figure 5 A typical learning curve plotting accuracy against number of data. The "knee point" indicates where improvements in accuracy with increases in data start to tail off (Smyth and Cunningham, 1996).

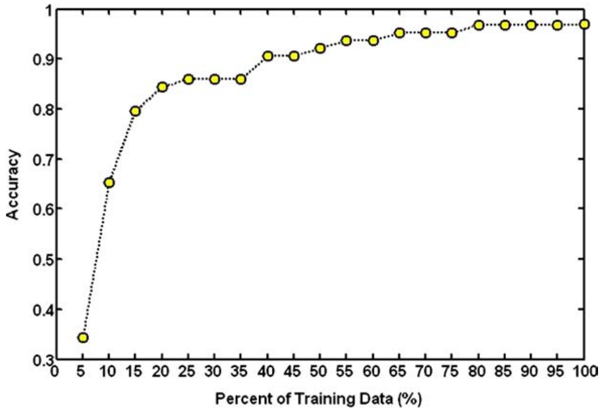
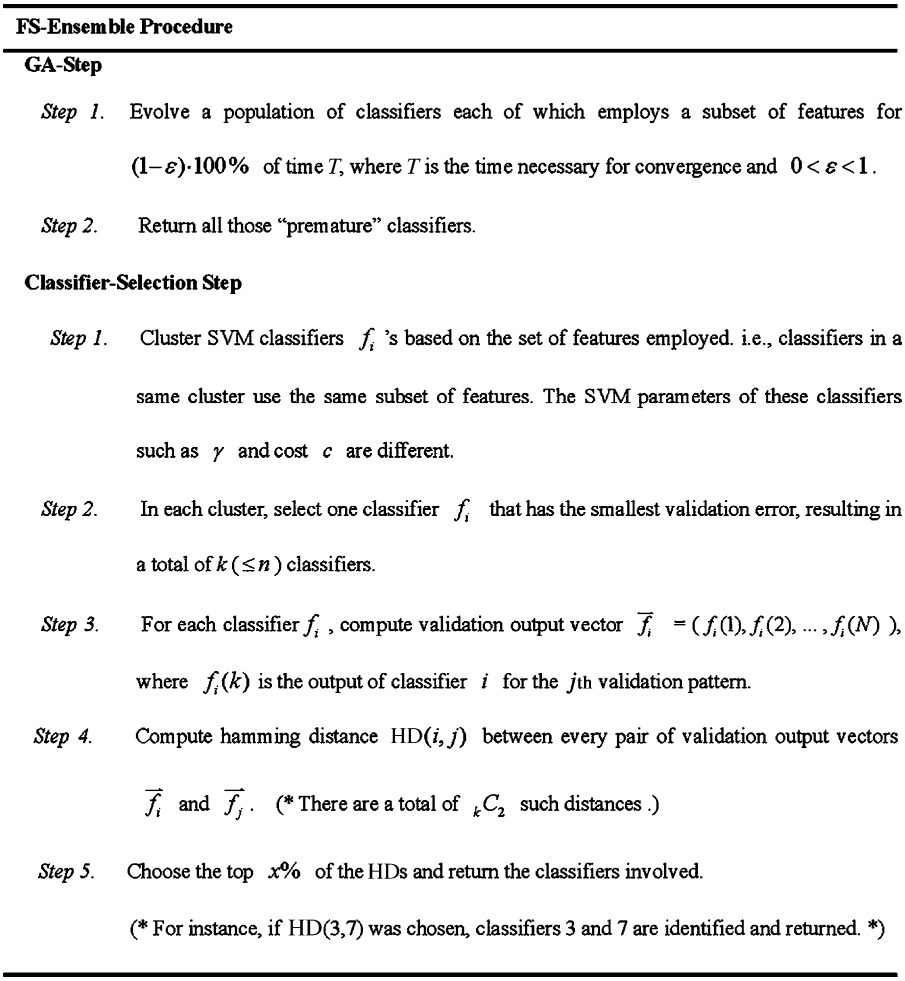


Figure 6 Experimental results plotting accuracy against number of data.

Recall that in section "GA—SVM wrapper approach for feature selection", the GA employed in the wrapper feature selection procedure deals with a population of SVMs employing different feature subsets. In the earlier stages of the GA evolution process, the candidate solutions or SVMs usually show a high level of diversity, while accuracy is low. On the other hand, in the later stages, the level of population diversity tends to decrease, while accuracy increases. Therefore, trade-off between diversity and accuracy can be made by selecting diverse candidates from the population immediately before it converges and then using them to create the ensemble. We propose an ensemble creation method based on this idea.

Previous researches on ensemble are based either on the difference of training sets, such as bagging (Breiman, 1996) and boosting (Shapire, 1990), or on difference of classifiers (Yao and Liu, 1998). The ensemble creation method proposed here is, on the other hand, based on the difference between feature subsets.

The procedure consists of 2 steps. First, in GA-step, those classifiers that were $(1-\varepsilon)\times 100\%$ evolved were picked. If they were evolved until convergence, they would be less diverse even though they would have better accuracies. Second, in the classifier selection step, among the classifiers employing the same set of features, only classifiers with the best validation performance are selected. Then their validation predictions are compared such that those classifiers with diverse prediction values are identified and used for the ensemble. The procedure is presented in Fig. 7. The problem of choosing the most diverse subset of classifiers based on their validation predictions can be transformed into a set packing problem, which is known as NP-complete (Karp, 1972). Thus, we employed a heuristic approach.



```
FS-Ensemble Procedure
GA-Step
  Step 1.  Evolve a population of classifiers each of which employs a subset of features for
           $(1-\varepsilon)\cdot 100\%$ of time $T$, where $T$ is the time necessary for convergence and $0<\varepsilon<1$.
  Step 2.  Return all those "premature" classifiers.
Classifier-Selection Step
  Step 1.  Cluster SVM classifiers $f_i$ 's based on the set of features employed. i.e., classifiers in a
           same cluster use the same subset of features. The SVM parameters of these classifiers
           such as $\gamma$ and cost $c$ are different.
  Step 2.  In each cluster, select one classifier $f_i$ that has the smallest validation error, resulting in
           a total of $k\,(\le n)$ classifiers.
  Step 3.  For each classifier $f_i$ , compute validation output vector $\overline{f_i}$ = ( $f_i(1), f_i(2), \ldots, f_i(N)$ ),
           where $f_i(k)$ is the output of classifier $i$ for the $j$th validation pattern.
  Step 4.  Compute hamming distance $\mathrm{HD}(i,j)$ between every pair of validation output vectors
           $\vec{f_i}$ and $\vec{f_j}$ . (* There are a total of $_kC_2$ such distances .)
  Step 5.  Choose the top $x\%$ of the HDs and return the classifiers involved.
           (* For instance, if HD(3,7) was chosen, classifiers 3 and 7 are identified and returned. *)
```

Figure 7 FS-Ensemble creation method.

Experimental results

Data collection

A program was developed to measure keystroke duration times and interval times in an X window environment on a Sun Sparc-station in our laboratory, and the data were collected via the keyboard connected to this workstation.

The data for both the owners and the imposters were collected for each of the 21 passwords, whose length ranges from 6 to 10. The keystroke duration and interval times were captured at the accuracy of milliseconds (ms). Each participant was asked to type his or her password 150—400 times. As for the novelty data, 15 imposters were given passwords beforehand, asked to practice typing these passwords, and then type them 5 times each, resulting in 75 impostor timing vectors for each password. We call those imposters "*imposters with practice*". The owners also provided a separate set of test patterns. Thus, we have 2 sets of 75 patterns, each representing novelty and normal state, respectively. Figs. 8 and 9 illustrate timing vectors of a certain password for the owner and imposters, respectively.

The first column of Table 1 (see section "Feature subset selection using GA—SVM wrapper") shows the collected passwords. One class of them includes common English phrases while another class includes combinations of abbreviated words or phrases, and some contain special characters, e.g., the password No. 21. Still another class includes passwords written in Hangul, the Korean alphabet, e.g., password Nos. 5, 6, and 8. We simply captured and showed the corresponding English characters, with regard to their positions on the keyboard.

Performance of SVM novelty detector

The SVM novelty detector with *Gaussian* kernel was trained and evaluated in terms of accuracy and learning time. The model reported an average error rate of 0.81%, with the worst one of 3.64%, and 8 perfect classifications. To compare its performance with that of previously proposed neural networks, both 2-layer AaMLP and 4-layer AaMLP were also trained and evaluated.

The 4-layer AaMLP achieved a performance similar to that of the SVM, with an average error rate of 1.21%, the worst error rate of 4%, and 8 perfect classifications. The 2-layer AaMLP, known to be linear thus limited (Hwang and Cho, 1999), showed a much worse performance, with an average error rate of 5.7%, the worst error rate of 17% with no perfect classifications. Fig. 10 illustrates the performances of the respective models. Both neural network models were trained with *Resilient backpropagation* algorithm, with a learning rate of 0.1 and a momentum term of 0.25.

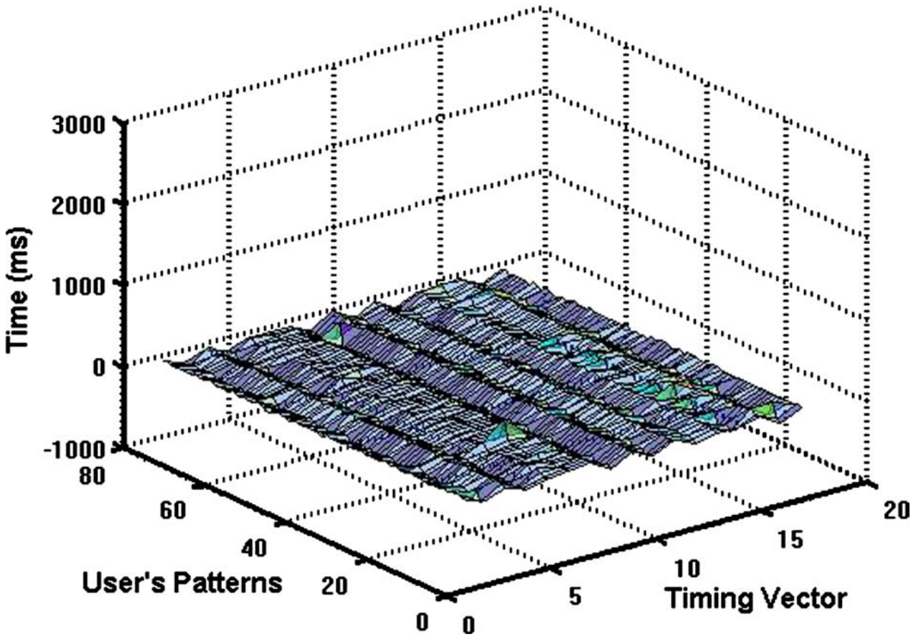


Figure 8 Example of user's patterns.

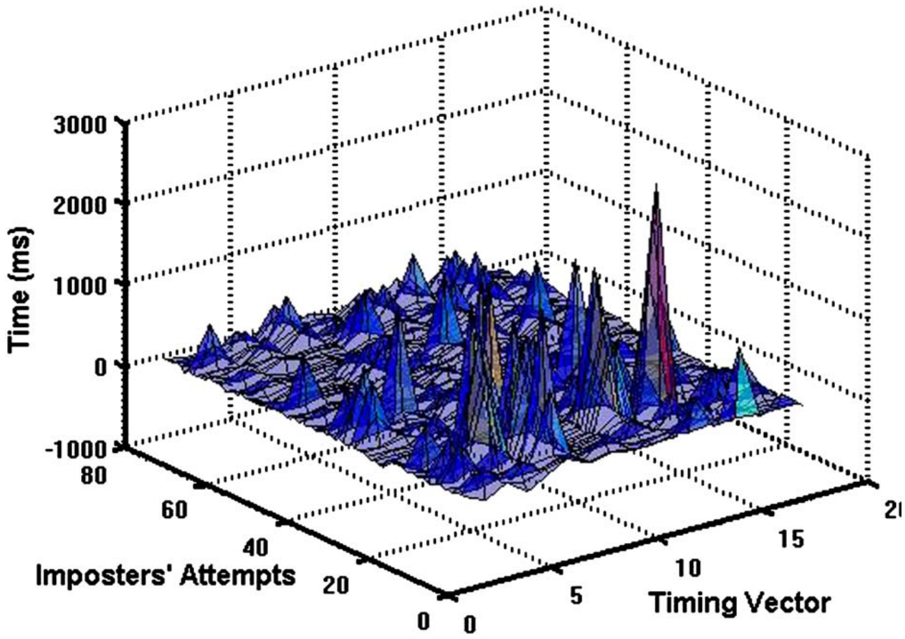


Figure 9 Example of imposter's patterns.

Table 1 FRR when FAR = 0 before- and after-FSS with the dimension used displayed inside parentheses

Owner ID	FRR (%)	
	Before-FSS	After-FSS
loveis.	19.32 (15)	2.68 (5)
i love 3	19.75 (17)	4.39 (4)
90200jdg	14.87 (17)	4.55 (4)
autumnman	19.82 (19)	3.60 (5)
tjddmswjd	19.88 (19)	4.23 (7)
dhfpql.	15.15 (13)	3.25 (7)
love wjd	19.80 (17)	1.25 (11)
ahrfus8	20.26 (15)	3.45 (6)
dusru427	14.85 (17)	3.25 (7)
manseiii	5.30 (17)	1.97 (6)
drizzle	12.40 (15)	4.35 (4)
beaupowe	10.53 (17)	4.68 (5)
tmdwnsl1	10.14 (17)	4.61 (6)
yuhwa1kk	12.79 (17)	3.49 (8)
anehwksu	12.62 (17)	3.18 (7)
rhkdwo	11.71 (13)	4.39 (4)
rlasus	13.29 (13)	2.81 (5)
dlfjs wp	19.74 (17)	3.95 (5)
dltjdgml	19.44 (17)	2.78 (4)
dirdhfmw	20.38 (17)	3.85 (6)
c.s.93/ksy 8	19.26 (21)	3.71 (8)
Minimum	5.30 (13)	1.25 (4)
Maximum	20.38 (21)	4.68 (11)
Average	15.78 (16.52)	3.54 (5.86)

In terms of the average learning time, SVM only took less than 0.1 s to construct a model, whereas 2-layer and 4-layer AaMLPs took 17 s and 124 s, respectively, as shown in Fig. 11.

In summary, the SVM achieved a performance comparable to 4-layer AaMLP but took much less time for training, i.e., less than 0.1% of the neural network model. These characteristics of the SVM clearly make it suitable to be used as a base learner in the wrapper feature selection approach.

Feature subset selection using GA—SVM wrapper

The proposed GA—SVM feature subset selection (FSS) was performed for all 21 password-typing patterns. The feature set dimensions were greatly reduced, 6 out of 17 features on the average (see Table 1).

In the case of ID "*atom*", for instance, the dimension was reduced from 15 to 5, with corresponding feature subset being "1001000 01100100". Only 3 keystroke durations and 2 keystroke intervals were selected. For each password, different features were selected.

Model accuracies were all improved with the selected feature subsets. The average FRR was reduced from 15.78% down to 3.54%. The minimum FRR was reduced from 5.5% to 1.25%, and the maximum FRR from 20.38% to 4.68%. In summary, the proposed feature subset selection of GA—SVM

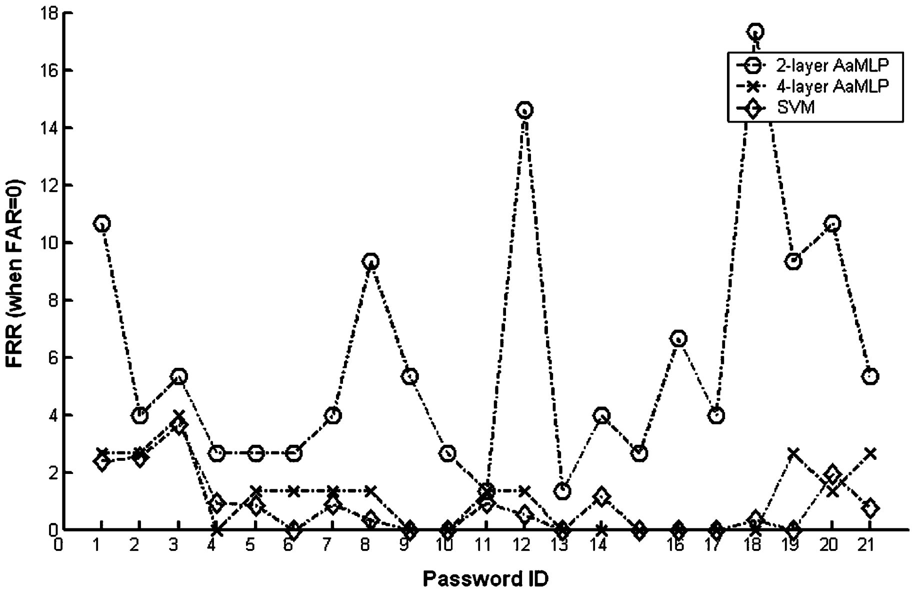


Figure 10 FRR for each password by 3 different models.

wrapper approach clearly improved the model performance.

Limited training patterns

Among the training patterns described in section "Data collection", only 50 timing vectors were sampled randomly for training, and the rest were used for testing. The 50 patterns were divided into 2 parts, i.e., 35 patterns for training and 15 patterns for validation. In the GA, the number of population was set as $n = 30$. The GA early stopping criterion was set as $\varepsilon = 0.2$, which was determined from an a priori experimental observation that it usually needed approximately 100 generations for the GA to converge, thus a population of "premature" solutions could be expected to appear at the point of 80% time. The threshold value on the hamming distances HD between each pair of validation output vectors was set as $x = 30\%$ since the threshold value $x = 30\%$ usually returned the desired number of diverse classifiers for ensemble creation, which was between 10 and 15.

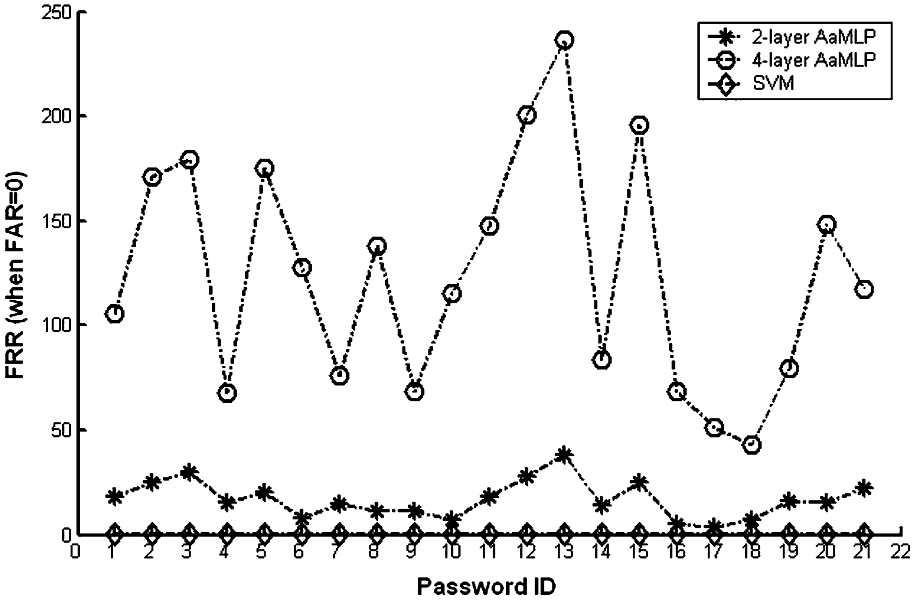


Figure 11 Learning time for each password by 3 different models.

Experiments were conducted to compare the effects of: (1) the proposed FS-Ensemble model, (2) the single SVM model using the full feature set (raw data), (3) the neural network model using the full feature set, (4) the ensemble model with randomly selected feature subsets (FR-Ensemble) proposed by Ho (1998), and (5) the SVM model trained with the best feature subset obtained by the GA—SVM wrapper. Table 2 and Fig. 12 show the FRR when FAR = 0 in the respective 5 models.

In model (1), the semi-converged classifiers that were chosen provided diversified and high quality committee members, thus they resulted in the best average FRR of 3.69%. Models (2) and (3) inevitably resulted in poor performance of an average FRR of 17.55% and 19.43%, respectively, since raw data contained noise or outliers, especially when the number of training patterns was limited. Model (4) presented more improved results (an average FRR of 8.29%) than models (2) and (3), but it presented results poorer than model (1). In model (5), the model implementing GA—SVM wrapper feature subset selection showed somewhat improved performance than that of models (2) and (3), i.e., an average FRR of 6.28%, but the performance is still not satisfactory because of the overfitting that resulted from limited training patterns. Furthermore, to show the advantage of the proposed FS-Ensemble model over other ones, we counted the number of passwords/data sets where a model gave the best performance in Table 2. For the 21 password IDs, the proposed FS-Ensemble performed the best. Models (4) and (5) did best in 1 and 3 times, respectively (a tied number 1 with model (1)).

Conclusion

In this article, we focused on some issues in keystroke dynamics identity verification that were raised from practical situations.

First, we applied an SVM novelty detector to keystroke dynamics identity verification and compared the performance with that of a neural

Table 2 FRR (%, when FAR = 0%) and the number of best results for various models using 50 training patterns

Password ID	Dimension	Models				
		FS-SVM ensemble	Single SVM with all features	Neural network with all features	Random feature ensemble (Hwang and Cho, 1999)	Single SVM with best selected feature
loveis.	15	1.32	15.63	15.63	5.24	7.69
i love 3	17	5.67	22.67	17.26	6.67	6.67
90200jdg	17	5.33	21.33	10.67	6.67	6.67
autumnman	19	4.67	14.67	9.33	10.67	7.33
tjddmswjd	19	5.67	10.67	9.33	12.67	9.33
dhfpql.	13	2.67	31.26	42.07	9.33	6.67
love wjd	17	1.33	13.33	9.33	5.24	2.67
ahrfus8	15	1.33	16.67	15.63	4.67	1.33
dusru427	17	4.67	24.67	31.26	13.33	5.24
manseiii	17	5.33	17.33	14.36	12.00	8.14
drizzle	15	0.00	9.33	25.33	8.67	2.67
beaupowe	17	5.40	9.33	10.67	10.67	9.33
tmdwnsl1	17	4.67	24.00	30.67	9.33	9.33
yuhwa1kk	17	1.33	17.33	24.36	9.33	8.67
anehwksu	17	5.33	28.00	18.67	5.33	7.33
rhkdwo	13	3.33	10.67	9.33	6.67	6.67
rlasus	13	3.67	14.67	23.67	7.33	6.67
dlfjs wp	17	5.14	17.33	34.67	5.14	5.14
dltjdgml	17	4.67	10.34	16.00	7.99	4.67
dirdhfmw	17	3.33	22.67	26.46	9.33	5.33
c.s.93/ksy 8	21	2.67	16.67	13.33	7.99	4.33
Minimum		0.00	9.33	9.33	4.67	1.33
Maximum		5.67	31.26	42.07	13.33	9.33
Average		3.69	17.55	19.43	8.29	6.28
Number of lowest FRR		21	0	0	1	3

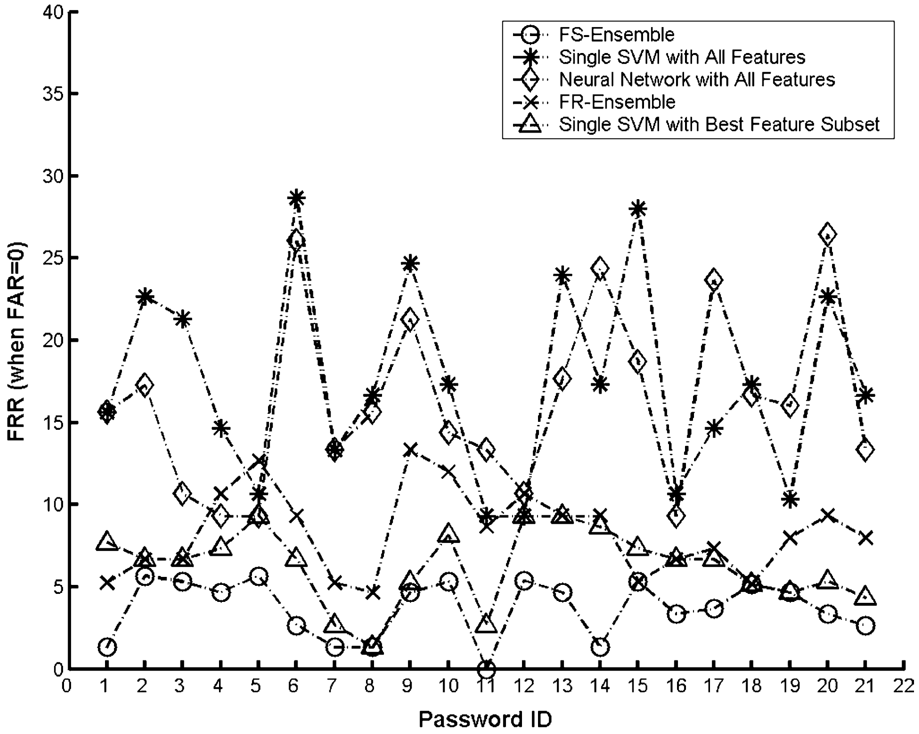


Figure 12 Performance comparison in respective situations with 50 training patterns.

network. The SVM and the 4-layer AaMLP showed a similar level of novelty detection performance. But the computational efficiency of the SVM is much higher than that of the neural network model: SVM models only need less than 0.1 s for training, whereas neural network models usually need more than 100 s to achieve a similar degree of accuracy.

Second, irrelevant or redundant features did not help in generalizing the discovered models, thus feature extraction was particularly important in practical problems. We proposed a GA—SVM-based wrapper approach for feature subset selection for keystroke dynamics identity verification. The SVM showed its excellence in both accuracy and learning speed, and it proved itself to be a suitable learner in the wrapper approach. The feature subset selection improved the model performance significantly.

Third, it is often the case in a biometrics-based identity verification system that the training data are insufficient. When a wrapper feature selection is applied, it often leads to overfitting. An FS-Ensemble was proposed to deal with such problems. Experiments were conducted to compare 5 models, i.e., (1) the proposed FS-Ensemble model, (2) the SVM model with full feature sets, (3) the neural network model with full feature sets, (4) the ensemble model with randomly selected feature subsets, and (5) the SVM model with single best solution obtained from the GA—SVM wrapper. Our experimental results show that the proposed approach has its advantages over other methods. With insufficient training data, models (2), (3) and (5) can hardly avoid overfitting, thus they result in low accuracy. Model (4) can alleviate the overfitting problem, but is somewhat unstable, because the committee members with diverse feature subspaces resulted from a random mechanism, and their quality cannot be guaranteed. Model (1), on the other hand, selects its committee members from the later stages of a GA-based feature selection procedure. Both the degree of diversity and quality are guaranteed, and thus they result in an improved model performance and stability.

There are still some limitations in our research. First, for the SVM novelty detector, the model hyper-parameters were tuned by way of *trial-and-error* method within a predefined range. This method limited the range of parameter values to a small set, and thus it was not efficient enough to find an optimal solution. Second, the GA was still a time consuming searching method although the fast learning speed of the SVM showed its fitness as an induction algorithm. Third, in the FS-Ensemble, the diversity of classifiers was simply measured using hamming distance for feature subset difference and classifier distance for learner diversity.

Our research efforts will focus further on fast searching algorithms that are suitable for parameter tuning and fast searching in a wrapper. Recent researches showed that the particle swarm optimizer (PSO) is very fast in finding an optimal solution with a performance comparable to that of GA. As for the proposed FS-Ensemble method, the diversity measurement, the stability and scalability of the model need to be studied further. Finally, since the purpose of this paper is to solve the problems that were encountered in the application of keystroke dynamics identity verification, no comparisons were made between the proposed method and other ensemble methods, such as bagging and boosting, which is another direction for future research.

Acknowledgements

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology.

References

Breiman L. Bagging predictors. Machine 1996;24(2):123—40.

Brown M, Rogers SJ. User identification via keystroke characteristics of typed names using neural networks. Int J Man Machine Stud 1993;39:999—1014.

Byun H, Lee S. Applications of support vector machines for pattern recognition: a survey. LNCS 2002;2388:213—36.

Cho S, Han C, Han D, Kim H. Web-based keystroke dynamics identity verification using neural network. J Organ Comput Electron Commerce 2000;10(4):295—307.

Dasarathy BV. Recognition under partial exposure and imperfect supervision. In: Proceedings of the International Conference on Cybernetics and Society; 1979. p. 218—21.

Gaines R, Lisowski W, Press S, Shapiro N. Authentication by keystroke timing: some preliminary results. Rand Report R-256-NSF. Rand Corporation; 1980.

Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison Wesley Publishing Company; 1989.

Ho TK. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Machine Intell 1998;20(8): 832—44.

Available from: http://www.biopassword.com.

Hwang B, Cho S. Characteristics of autoassociative MLP as a novelty detector. International Joint Conference on Neural Networks, Washington, DC, USA; 1999.

Karp RM. Reducibility among combinatorial problems. In: Miller RE, Thatcher JW, editors. Complexity of computer computations. New York: Plenum Press; 1972.

Leggett J, Williams G, Usnick M, Longnecker M. Dynamic identity verification via keystroke characteristics. Int J Man Machine Stud 1991;35:859—70.

Obaidat M, Sadoun S. Verification of computer users using keystroke dynamics. IEEE Trans Syst Man Cybernet Part B 1997;27(2):261—9.

Schölkopf B, Platt J, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87. Microsoft Research, Redmond, WA; 1999.

Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC. Support vector method for novelty detection. In: Solla SA, Leen TK, Müller K-R, editors. Advances in neural information processing systems, vol. 12. MIT Press; 2000. p. 582—8.

Shapire RE. The strength of weak learnability. Machine Learning 1990;5:197—227.

Smyth B, Cunningham P. The utility problem analysed: a case-based reasoning perspective. In: Smith I, Faltings B, editors. EWCBR'96 Advances in Case-Based Reasoning. Lecture Notes in Artificial Intelligence: Springer-Verlag; 1996. p. 392—9.

Vapnik V. The nature of statistical learning theory. New York: Springer-Verlag; 1995.

Yang J, Honavar V. 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; 1998.

Yao X, Liu Y. Making use of population information in evolutionary artificial neural networks. IEEE Trans Syst Man Cybernet Part B 1998;28(3):417—25.

Yu E, Cho S. Novelty detection approach for keystroke dynamics identity verification. Fourth International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2003 Hong Kong, China; 2003.

Enzhe Yu received B.S. in Computer Science & Engineering from Harbin Institute of Technology, China, in 1997, and M.S. in Industrial Engineering from Kangnung National University, Korea, in 2000. He is currently pursuing the Ph.D. degree in the Department of Industrial Engineering, College of Engineering, Seoul National University, Korea, with his major data mining. He has published a number of papers in international journals and conferences. His research interests include neural networks, evolutionary computing, data mining and computer security.

Sungzoon Cho is associate professor in the Department of Industrial Engineering, College of Engineering, Seoul National University. He received B.S. and M.S. in Industrial Engineering from Seoul National University. He also received M.S. in Computer Science from University of Washington and Ph.D. in Computer Science from University of Maryland. He published over 100 papers in the areas of neural networks, machine learning and data mining in such journals as IEEE Transactions on Neural Networks, Neural Computation, Pattern Recognition and Neural Processing Letters. He also holds a US patent concerned with keystroke-based user authentication.