Java-Based Internet Biometric Authentication System

Ross A.J. Everitt and Peter W. McOwan

Abstract—An online biometric verification system for use over the Internet and requiring no specialist equipment is presented. Combining two distinct tests to ensure authenticity, a typing style test and a mouse-based signature test, achieves a fraudulent access rate of ≈ 4.4 percent, while authentic users access with a rate of ≈ 99 percent.

Index Terms—Authentication, biometric, Internet, Java, keyboard dynamics, signature, verification.

1 Introduction

With the increasing use of the Internet as a business and social tool, it is becoming more important that secure access to sensitive and personal information can be provided. Biometrics, the application of statistical analysis to identify individuals through their biological or physiological characteristics, is emerging as a key aspect in new security systems. Using biometrics, it is possible to avoid pitfalls encountered with traditional security systems where users are required to keep information, such as passwords, safe [1].

Biometric data can be classified as physiological or behavioral [2]. Physiological data remains stable over time (barring injury), examples include fingerprints [3], iris and retinal scans [4], [5], and hand geometry measurements [6]. Behavioral data may change over time, typical examples include signatures [7], [8], [9], [10], voice prints [11], [12], and typing styles [13], [14]. Among these methods signatures are most likely to be subject to active forgeries.

In this paper, we present an online behavioral biometric verification system that is used in addition to the standard password match. The biometric system improves upon the security level provided by password matching while greatly reducing the risk of dictionary-based attacks. The system can be customized to multiple Internet-based applications requiring secure authentication. The system uses no specialized equipment, requiring only an Internet capable computer with a keyboard, a mouse, and a Java compliant browser; other systems require specialist equipment such as scanners (e.g., fingerprint, iris, retinal) and microphones.

2 System Overview

The system introduced in this paper uses a hybrid approach to verification, requiring an authenticity confirmation from both a typing style test and a signature match. The first stage of our system uses a neural network to verify authenticity based upon keyboard dynamics of password input [15], [16]; the pattern of button presses as the user enters a password. Using the password input as the source of biometric data means the security benefits of standard password verification are enhanced, while placing no increase upon the user's cognitive load.

The second stage of the system uses additional neural networks that verify authenticity based upon an online signature match using both temporal and spatial information. Unlike other systems [17], [7], [8], [18], [19], [20], [21], [22], [23], we use the mouse as the input device. Two existing user skills are therefore built upon; mouse use and signature writing. Although a pen-based system could be more desirable for ease-of-use this would negate the possibility of accessing the system via the Internet using no specialized hardware; a key advantage of the system presented here. Fig. 1a shows that complex signatures may be drawn using the mouse.

By using passwords and signatures to gather biometric data, we avoid negative social stigmas, unlike, for example, fingerprint systems. In a questionnaire-based study conducted with 35 participants, we determined that 83 percent of people are happy to provide signatures as a means of verification and of these 97 percent would be happy to provide a signature for use on the Internet.

2.1 System Function

The neural network-based system functions in three distinct modes: registration, training, and verification. During the registration phase, new users are required to select a username and input a chosen password and signature multiple times (40 in our experiments). The gathered biometric data is processed to extract salient information. The details of the salient information, specifically, the feature points used for the authentic user are then stored in a template file. During the training phase, a novel technique is used to generate forged samples automatically from authentic user samples. These forged samples, together with the authentic user samples, are provided to a back-propagation neural network, which is trained and stored upon the server. During verification, the user logs into the system via an applet that accepts a username, password, and signature. The user template file, retrieved from the server, contains details of the authentic user's salient features; these features are then extracted from the input biometric data and sent to the server for neural network verification. Because the template only contains details of which features to use, and not the values that they should produce, no clue, neither explicit nor implicit, as to signature shape is provided. The template may therefore be communicated safely between client and server.

2.2 Data Acquisition

Initial data was gathered from an experiment involving 41 participants between the ages of 20 and 30, with each participant registering with the system via a Web-based applet from various (uncontrolled) locations. Each of the 41 participants was requested to register using their own details and also asked to provide a test set of forged samples for a random selection of other users. This allowed an assessment of the authentic user consistency levels and the degree of variation between authentic and impostor samples, to determine uniqueness.

To be truly portable, this system makes minimal assumptions about the available technology at the point of use. The main issues encountered during data analysis relate to the direct reliance upon the local Java VM (Virtual Machine) to accurately sample information. At present, the Java VM technology produces inconsistent and nonuniform sampling rates due to its reliance upon the underlying hardware and software of the client machine. This inconsistency means that noise is introduced into any sampled data. This system has therefore been designed to be tolerant of this noise by using neural networks trained with noisy genuine data and appropriately noisy automatically generated forged data samples.

2.3 Preprocessing

No two signatures are identical, even when signed by the same person. The lengths of the signature trace (in terms of the total number of sampled points N), the spatial size and temporal information will all vary. The input signature trace therefore needs to be preprocessed to reduce the effect of these differences and to convert it into a standard format. Signature traces are preprocessed to normalize the total arc length (disjoint segments are joined to produce a single continuous arc). Next, the total time taken to produce the trace is normalized. Finally, the traces are linearly time warped so that the N points contained in the signature trace are replaced by N′ temporally equidistant points using the process

• *The authors are with the Department of Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK. E-mail: pmco@dcs.qmul.ac.uk.*

Manuscript received 14 Feb. 2002; revised 5 Aug. 2002; accepted 9 Aug. 2003. Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115896.

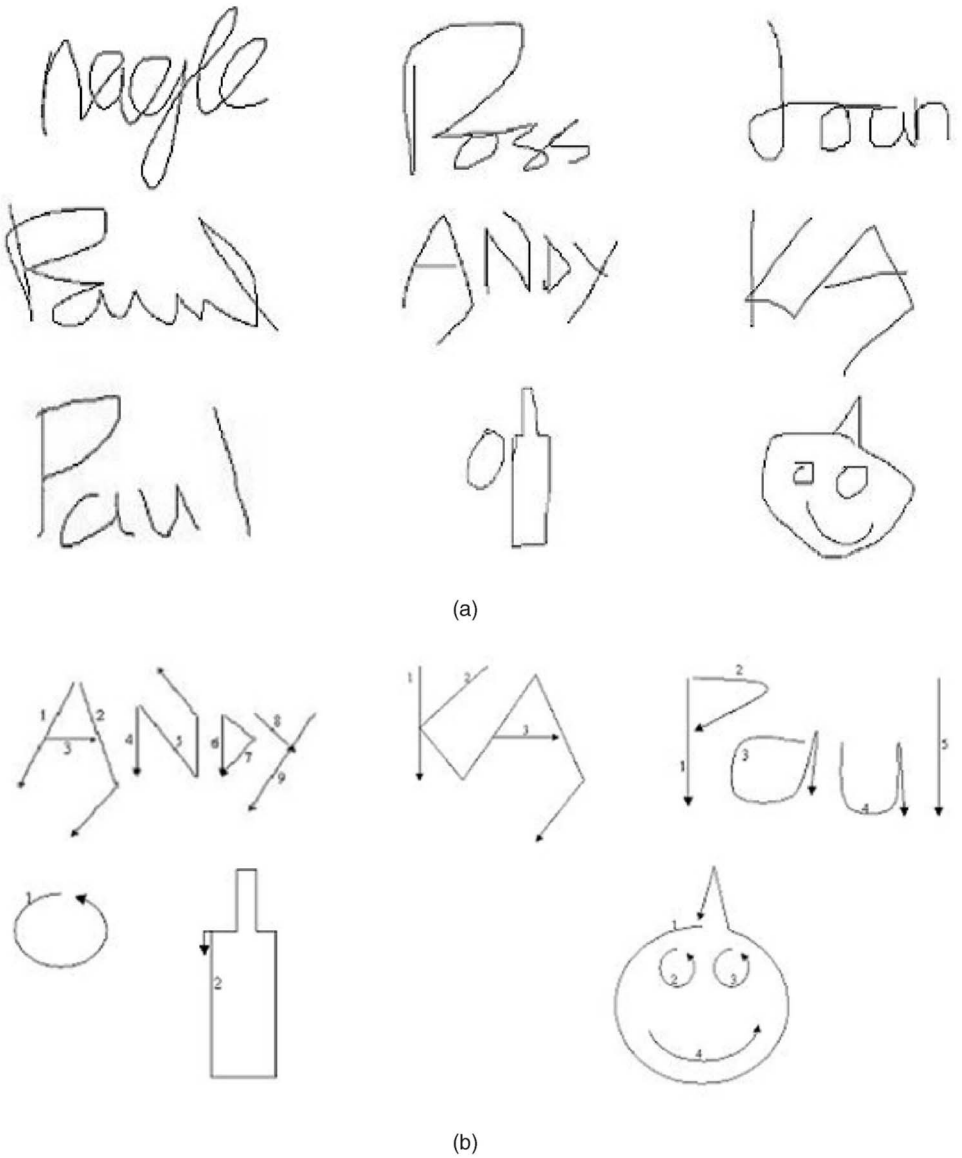

Fig. 1. Typical signatures used in the system. The images above represent typical signatures supplied by the users of our system and manually constructed schematics showing stroke sequences used in later testing of the system with forged signatures. The images show that remarkably complex signatures may be input with a mouse. (a) The images show single stroke signatures (first four), multistroke signatures (next three), and picture signatures (remaining two). (b) These images show the sequence of strokes as input by the authentic user.

described by Lee [18]. This step is necessary because the Java VM is unable to sample at a constant rate, thus adding excess noise to the input signature and exaggerating differences in input.

3 Information Extraction

From the keyboard dynamics of password input, we extract two complete sets of biometric data; latency times and hold times. Hold times represent the length of time each key is held down, while latency times indicate the time from releasing one key until pressing the next. Because these values are to be fed into a neural network for training they are normalized by the total time (to type the password) to prevent network weight saturation during training.

It is possible to represent a signature using all information obtainable from the raw signature trace. This is, however, undesirable because much of the data will not provide a significant degree of uniqueness or consistency. The usage of such information could, therefore, prove to be counterproductive. Storing all of the information is also costly (in terms of space) and has implications for processing overheads when training networks and verifying signatures.

A signature may be represented by a set of extracted features rather than all of the raw data. This system adapts a technique used by Ozcan and Mohan [24], [25] to perform partial spatial shape matching. Using this technique, angle and distance relationships between internal points, as defined by (3.1), (3.2), and (3.3) and illustrated by Figs. 2a, 2b, 2c, and 2d, are used for representation.

The Euclidean distance, d_{ij}, is between two points $S_i = (x_i, y_i)$ and $S_j = (x_j, y_j)$. N' is the number of temporally equidistant points contained in the signature after performing linear time warping.

$$d_{ij} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2}, \quad \text{where } [(0 \le i < j \le N') \vee (i = N', j = 0)]N' = 100. \tag{3.1}$$

The vector associated with a point, S_i, is obtained by the function $V(S_i)$, which returns either a vector from the previous point to the current, or from the last point to the first in the trace.

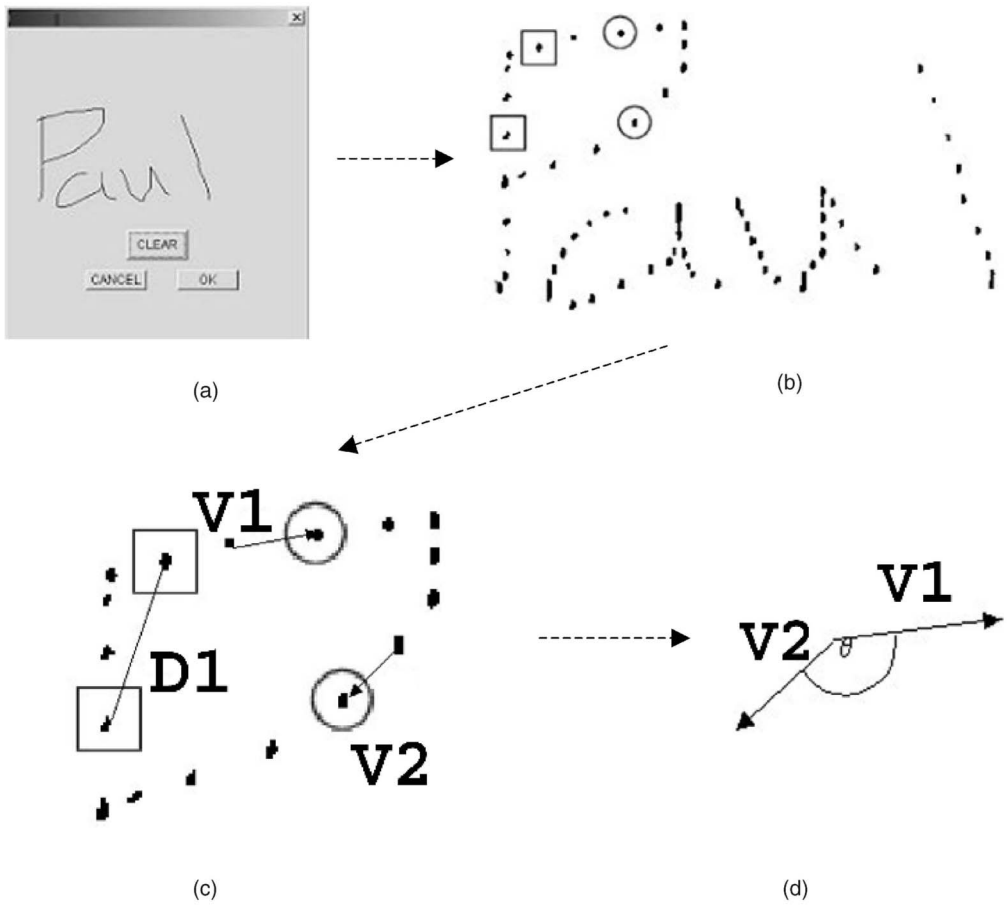

Fig. 2. Extracting relationships from signatures. The above images show how relationships are extracted from an input signature. (a) The input signature as seen by the user. (b) The input signature trace represented as sampled points; the circles indicate the two points from which an angle is to be extracted, while the squares indicate the points from which distance information is to be extracted. (c) An expanded view of the region information is to be extracted from. D1 indicates the Euclidean distance vector between the two points. The two vectors V1 and V2 show the vectors that represent the two points. (d) The clockwise angle between the vectors V1 and V2 is determined.

$$\begin{array}{ll} V(S_i) = \overline{S_i - S_{i-1}}, & \text{if i} > 0 \\ V(S_i) = \overline{S_i - S_{N'-1}}, & \text{otherwise where } \overline{S} \text{ indicates} \\ & \text{vector normalization.} \end{array} \quad (3.2)$$

The angle, a_{ij}, from $S_i$to S_j,where $0 \leq i, j \leq N'$ is obtained from the function A, which returns the clockwise angle between the two vectors (see Fig. 2d):

$$a_{ij} = A(V(S_i), V(S_j)) \text{ where V(S}_\text{x}\text{) is defined by (3.2).} \quad (3.3)$$

To use the extracted angle and distance information to characterize a signature trace, a technique must be implemented to obtain both the salient angle and distance relationships from any input signature. The technique must obtain two sets of signature points (one angle set and one distance set) from which relationships can be extracted. This is performed with the intention of minimizing within-class variance and maximizing between-class variance, where within-class variance is the degree to which patterns belonging to the same class differ and between-class variance is the degree to which patterns belonging to different classes differ. Two approaches used to extract this information are now presented.

3.1 Ranking Approach

Initial tests used a simple ranking algorithm to extract the most consistent relationships within the data. The fitness for a relationship between two points S_i and S_j is given using the standard deviation equation defined by (3.4).

$$\text{fitness(S}_\text{i}\text{, S}_\text{j}\text{)} = \sqrt{\left(n \sum_{k=0}^{n} f(x_k)^2 - \left(\sum_{k=0}^{n} f(x_k) \right)^2 \right) \Big/ n(n-1)}, \quad (3.4)$$

where n is the number of signatures input by a specific user during registration and $f(x_k)$ is a function that calculates either the distance d_{ij} (3.1) or angle a_{ij} (3.3) between the points S_i and S_j (as defined by (3.1), (3.2), and (3.3) for signature k.

Relationships are ranked in order of fitness and the 10 fittest angle and 10 fittest distance relationships used for signature representation. Using this technique, the relationships all contain two points that are too close together, in a temporal sense, to provide an acceptable level of between-class variance. This occurs because verification relies too heavily upon spatial matches, which proved less reliable in tests than spatio-temporal matches.

In order to increase the amount of between-class variance when obtaining relationships, the fitness function defined by (3.4) was redefined by (3.5) so that a bias is applied toward points that are further apart temporally.

$$\begin{aligned} \text{fitness(S}_\text{i}\text{, S}_\text{j}\text{)} = & \sqrt{\left(n \sum_{k=0}^{n} f(x_k)^2 - \left(\sum_{k=0}^{n} f(x_k) \right)^2 \right) \Big/ n(n-1)} \\ & - (|j-i|/N'), \end{aligned} \quad (3.5)$$

where |j − i| is the modulus of j − i. N' is defined by (3.1) and $f(x)$ and n are defined by (3.4).

The result of redefining (3.4) is that the identified relationships provide a higher amount of between-class variance, as illustrated

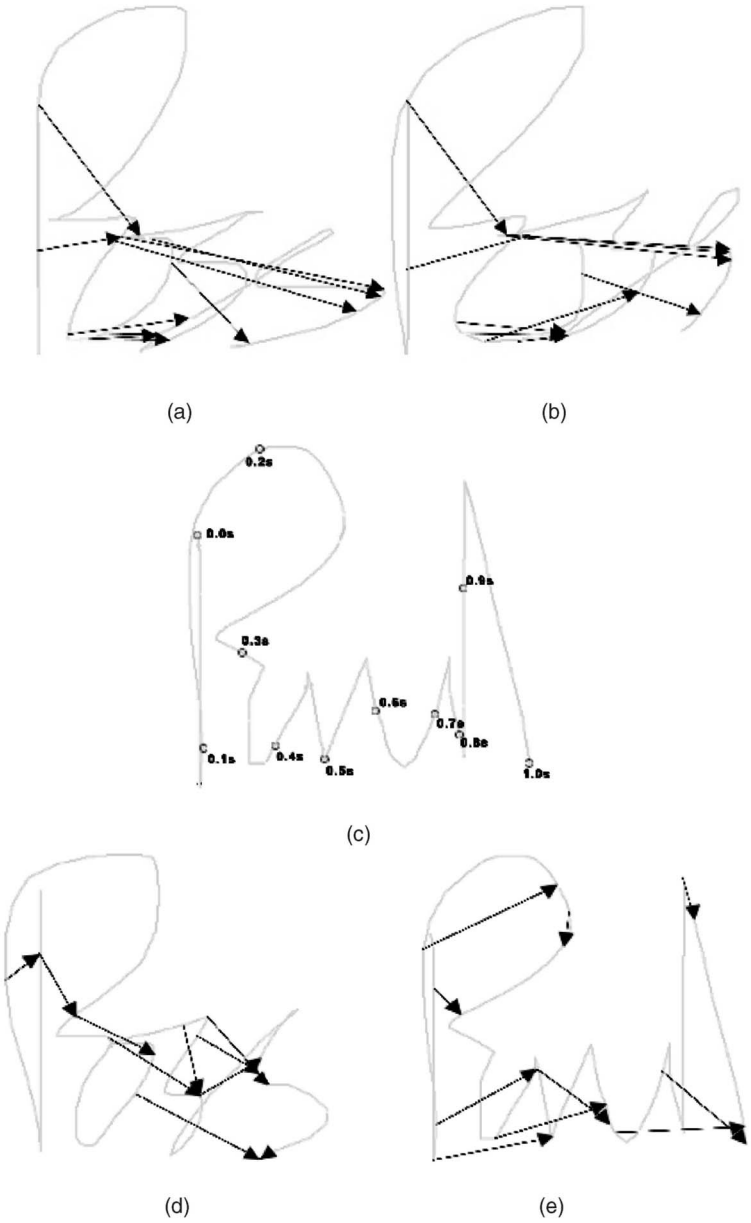

Fig. 3. Information reduction in signatures. The signatures above have been input into the system during the registration phase. In all cases, the arrows represent the distance relationships (the arrow heads show the *to* points). (a) and (b) The relationships extracted using the ranking approach form localized groups placing a high reliance upon a small number of comparison areas. (c) This diagram shows a signature split into 10 temporally equidistant sections. When using genetic extraction a bonus will be awarded for each temporal section containing a *from* point (start of an arrow) and *to* point (end of an arrow). (d) and (e) The arrows show that the genetically determined relationships are more evenly distributed throughout the signature than using the ranking approach.

by Figs. 3a and 3b. The relationships do, however, form localized groups meaning verification is performed with a high reliance upon a small number of comparison areas. This means that only a small number of areas need to be replicated for authenticity to be confirmed. This information is redundant, in any case, because the neural networks will implicitly model areas of high consistency.

3.2 Genetic Approach

The novel genetic algorithm (GA) described in this section improves upon the ranking approach to relationship extraction. The GA finds relationships possessing better within and between-class variance values than the ranking approach, using standard survival of the fittest, selection, crossover, and mutation [26]. Relationships are found that are well distributed throughout the signature, as illustrated by Figs. 3d and 3e. The GA is used to encode each individual user's angle and distance relationships and is defined by (3.6), (3.7), and (3.8).

The genetic population P consists of c individual chromosomes (100 in our experiments):

$$\mathrm{P} = (\mathrm{C}_1, \mathrm{C}_2, \ldots . \mathrm{C}_\mathrm{c}). \tag{3.6}$$

Each chromosome C consists of g genes (10 in our experiments):

$$\mathrm{C} = (\mathrm{G}_1, \mathrm{G}_2, \ldots . \mathrm{G}_\mathrm{g}). \tag{3.7}$$

Each gene G contains two different points in the signature, i.e., a single relationship.

$$\mathrm{G} = (\mathrm{S}_\mathrm{i}, \mathrm{S}_\mathrm{j}) \quad \text{where i} < \mathrm{j}. \tag{3.8}$$

The fitness of the individual genes is determined by the fitness function defined in (3.5). Our Chromosome fitness function, CF, defined by (3.9) averages the fitness of each of the constituent genes where α, β, and χ are bonus functions described below:

$$CF = \sum_{i=1}^{g} fitness(G_i) \Big/ g + \alpha + \beta + \chi, \text{where CF } g \text{ is defined by (3.7) and } fitness(G_i) \text{ is defined by (3.5) and (3.8).} \quad (3.9)$$

To determine the bonus values, the signature must first be divided into g sections. The bonus functions may then be used to satisfy a set of conditions. Described below are the conditions necessary to extract salient features and the bonus functions used to achieve this.

1. Extracted relationships provide low levels of within-class variance, as they capture the salient features of the user signature. The low level of within class variance is provided by the use of the formula $fitness(G_i)$, which determines the fitness of individual relationships within the preprocessed signature trace.
2. The relationships are well-distributed throughout the signature, maximizing the likelihood of high between-class variance. This is achieved using functions β and χ, which both lie in the range {0..1}. Function β returns a bonus of $1/g$ for each section containing a *from* point (S_i in $\mathrm{G} = (\mathrm{S_i}, \mathrm{S_j})$). Function χ returns a bonus of $1/g$ for each section containing a *to* point (S_j in $\mathrm{G} = (\mathrm{S_i}, \mathrm{S_j})$).
3. The chromosome contains consistently fit genes, rather than some fit and some unfit. This prevents authenticity being verified based upon inconsistent (unfit) relationships. Function α achieves this goal by penalizing chromosome fitness in proportion to the fitness of the worst gene, using the function $(\mathrm{Min}(1.0 - \text{standard deviation}(\mathrm{G_{0..i}})))$.

4 Learning

Any neural network-based verification system requires sufficient training data to enable generalization [27]. The authentic user may provide positive samples at the registration phase. There are, however, two main problems associated with obtaining false data from real people; the authentic user's password and signature must be made available to others and people willing to provide a sufficient number of good quality forged samples must be found.

4.1 Profile Space, Boundary Space, and False Pattern Generation

The biometric data obtained from a specific user input resides in a small sub-space of the entire signature space. It is possible to authenticate an individual based upon whether their input data falls within this *profile space* region; the difficulty for any system, therefore, is determining the size and shape of the authentic user's profile space. In this system, the means and standard deviations extracted from the data supplied at the time of registration (hold/latency times or angle/distance relationships) are used to provide an approximate model of the profile space. In each dimensional plane, we assume a circular distribution; the mean values provide the profile space center, while standard deviation values provide the radius.

The most difficult forgeries to recognize are those that are very similar to authentic samples, lying close to the authentic user's profile space. Forgeries which reside further from the profile space can more easily be rejected by a verification system and need fewer training examples. In training a high proportion of false samples lying close to the profile space boundaries, within a *boundary space region* were used along with a few outlying samples to ensure a correct modeling of the problem domain. The boundary space region is an enclosing subspace (around the profile space region) whose radius is the same as the profile space radius (in each dimension) and extends a further distance of 0.25 times the relevant radius. False samples are generated within the boundary space using pseudorandom values for each axis, based upon the authentic user's characteristic patterns (additional true samples are similarly generated within the profile space using the same technique). Difficult forgeries are generated because they often lie outside the profile space in only one dimension.

4.2 Network Training

To perform user verification, this system uses three neural networks each trained using the back-propagation algorithm [27]. The first network uses hold and latency times to test typing style, the second and third networks use angle and distance information, respectively, to test the input signature. Separate angle and distance networks are used because a combined network could not be trained to correctly model the problem. In order for authenticity to be verified as genuine, the user input data must pass all three tests.

When performing gradient descent on the networks, it is possible to overfit a problem such that the network remembers the input patterns rather than establishing an ability to generalize. The global minima of the training data error surface may provide a bad solution here because the input patterns are remembered. In order to combat this problem, this system uses a validation set during training to test for an ability to generalize; gradient descent is performed with respect to the training set but the previously unseen validation set is used to test for generalization ability. To create the training, validation, and testing sets the authentic user data (and autogenerated true data) is split between the three sets. False data is generated for each using boundary space generation, with the validation and testing sets using a boundary space slightly closer to the profile space than the training set so that performance and ability to generalize is assessed based upon more difficult samples. The proportion of data in the training, validation, and testing sets, respectively, is (3 percent, 27 percent, 70 percent), (15 percent, 0 percent, 85 percent), and (4 percent, 0 percent, 96 percent) in the format (true, true (autogenerated), false (autogenerated)).

The input layer node size for the angle network and the distance network is set at *10* and for the keyboard dynamics' network is set at ((number of letters in password*2) – 1). The output size for each of the networks is 1 (accept or reject supplied credentials). To determine the hidden layer size and structure for the networks, we used a genetic algorithm employing an adaptive multipoint crossover strategy combined with a survival of the fittest mechanism, roulette wheel selection, and mutation operator [26]. The GA indicated that a single hidden 20-node layer should be used for the password network. For the angle and distance networks, the GA indicated two hidden layers, of 30-nodes each, should be used. The network threshold is set to provide a good balance between FAR and FRR (with respect to the testing set). We select the i value that maximizes $(1 - FAR_i^2 + FRR_i^2)$, where FAR_i and FRR_i are the FAR and FRR values obtained when the network is run with threshold i.

5 Results

In this section, the results obtained from this system are presented in terms of FAR (False Accept Rate) and FRR (False Reject Rate). FAR is the rate at which forged samples are accepted as genuine and FRR is the rate at which genuine samples are rejected as forgeries.

5.1 Experimental Methods

To obtain FRR information users were asked to log into the system twice a day for two weeks; at each login session, the users were allowed five attempts to access the system, if after five attempts their login was unsuccessful, the session was classed as a false reject. This methodology is the same as that used by Zwiesele et al. [28].

FAR information was obtained from a group of 41 participants, each of whom was given a copy of the authentic user's login, password, signature, and stroke sequence (the details of how the signature was written—see Fig. 1b). None of these would be available to genuine forgers. The participants were asked to attempt to fraudulently access the authentic user's account using these details, and if successful, a false accept was recorded. The experiment provided a total of 1,500 fraudulent attempts to access the 41 genuine accounts.

5.2 System Performance

The FAR results obtained using this system are hereby presented; values for 95 percent confidence intervals are provided in brackets. The system has an overall FAR of 8.5 percent (±4.2 percent) when signature stroke sequences are supplied (see Fig. 1b) and 7.0 percent (±3.9 percent) if they are not. Using a familiar password and a single or multistroke signature (which users can fairly easily generate), users can expect an FAR of 4.9 percent (±4.1 percent) if stroke sequences are supplied to forgers and 4.4 percent (±3.8 percent) if they are not. Using picture signatures (which authentic users found hard to produce consistently) the FAR increases to 16.3 percent (±11.4 percent) if signature stroke sequences are supplied and 12.1 percent (±10.9 percent) if they are not.

A FRR was recorded for the 41 users who logged into the system 20 times over two weeks. This rate varies little irrespective of signature type. The overall FRR produced by this system degrades as follows: FRR(5) = 0.2 percent, FRR(4) = 2.4 percent, FRR(3) = 7.8 percent, FRR(2) = 21.2 percent, and FRR(1) = 38.6 percent, where FRR(1) means FRR given one attempt at each login.

6 Conclusions

This paper has introduced a new approach for providing secure access over the Internet using biometric verification. The system uses a hybrid test to ensure that the set of credentials supplied to the system at the login stage are genuine. The system is novel because it is, to the best of the authors' knowledge, the only mouse-based Internet signature verification system to be developed at this point in time. The system is specifically designed for use in a potentially hostile real world environment with uncontrolled and nonstandard equipment.

The FAR of 4.4 percent produced by this system is extremely encouraging and compares well with figures cited by other specialist technology pen-based systems including Higashino (0.61 percent) [10], Martens (1.5 percent) [15], and Hesketh (3.6 percent) [9]. The keystroke-based system developed by Robinson [10] quotes an FAR of 9 percent. In our system, the FAR for an average user is largely determined by the type of the password and signature chosen, a good addition to this system may therefore be a complexity measure of the input signature that provides an estimation of security level fed back to the user at the time of registration. This would assist the user in choosing an appropriate form of their signature for the desired level of security.

The FAR provided by this system is affected by the type of password chosen by the authentic user (familiar versus randomly selected). The use of a familiar password provides higher between-class variance and lower within-class variance because users have developed a style and level of consistency of input; however, when using a randomly assigned word the input is less fluid and well practiced and therefore less unique. This observation highlights the importance of providing a system that may adapt over time to model the improvements in user consistency. Adaptation can be provided by performing periodic retraining of the entire system using data accumulated from successful logins. Alternatively data could be presented to the trained networks at each successful login from which an output error is calculated. A single back-propagation pass is then performed, allowing gradual evolution of the networks.

The FRR(5), where FRR(5) means FRR given five attempts at each login, of 0.2 percent provided by this system compares well with the systems tested by Zwiesele et al. [28] where rates of 65.4 percent and 14.5 percent are cited. This system also compares favorably when fewer than five attempts are allowed at each login.

It is anticipated that the FAR and FRR will decrease as better hardware (improving sampling consistency) becomes available. We believe that by producing forged samples based upon the generation of signature traces rather than relationship values the rates could be reduced even further. This technique could provide signatures that look like the authentic signatures and so relationships could then be extracted based upon within and between-class variance rather than using empirical values. We also note that this system allows the threshold to be changed so that greater security or reduced verification levels are required to allow users into the system, this is useful as different applications will provide different security requirements.

The results produced by this system are extremely encouraging and suggest that widespread implementation of the system over heterogeneous networks should be further investigated. The system produces good FAR and FRR values even when genuine passwords, signatures, and stroke sequences are provided to forgers. Forgers cannot obtain these details from the final system because only relationships and not the entire signature trace are stored. The system is likely to benefit from lower skepticism levels than other biometric systems because familiar tasks are built upon, thus meaning that a faster deployment time should be possible.

Acknowledgments

The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) whose funding made this project possible. Thanks should go to Dr. Janak Sodha for the information provided upon training neural networks with the use of a validation set. The authors would also like to thank Andrew Anderson for the invaluable insight he provided while proof reading draft copies of this paper and to Christopher Bush for editing the images published in this paper. The system described in this paper is the subject of patent protection.

References

[1] R. Clarke, "Human Identification in Information Systems: Management Challenges and Public Policy Issues," *Information Technology and People*, vol. 7, no. 4, pp. 6-37, Dec. 1994.
[2] B. Miller, "Vital Signs of Identity," *IEEE Spectrum*, vol. 31, no. 2, pp. 22-30, Feb. 1994.
[3] A. Roddy and J. Stosz, "Fingerprint Features—Statistical Analysis and System Performance Estimates," *Proc. IEEE*, vol. 85, no. 9, pp. 1390-1422, Sept. 1997.
[4] S. Gordon, "Ocular Biometrics: For Your Eyes Only," *Opto & Laser Europe*, no. 84, May 2001.
[5] Y. Zhu, T. Tan, and Y. Wang, "Biometric Personal Identification Based on Iris Patterns," *Proc. Int'l Conf. Pattern Recognition*, vol. 2, pp. 805-808, 2000.
[6] R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos, "Biometric Identification through Hand Geometry Measurements," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1168-1171, Oct. 2000.
[7] S. Hangai, S. Yamanaka, and T. Hamamoto, "On-Line Signature Verification Based on Altitude and Direction of Pen Movement," *Proc. IEEE Int'l Conf. Multimedia and Expo*, vol. 1, pp. 489-492, 2000.
[8] B. Herbst and D. Richards, "On an Automated Signature Verification System," *Proc.IEEE Int'l Symp. Industrial Electronics*, vol. 2, pp. 600-604, July 1998.
[9] G.B. Hesketh, "Countermatch: A Neural Network Approach to Automatic Signature Verification," *Proc. IEE Colloquium on Neural Networks for Industrial Applications*, pp. 2/1-2/2, Feb. 1997.
[10] J. Higashino, "Signature Verification System on Neuro-Computer," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, vol. III-C, pp. 517-521, 1992.
[11] T. Clarkson et al., "Speaker Identification for Security Systems Using Reinforcement-Trained pRAM," *IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 31, no. 1, pp. 65-76, Feb. 2001.
[12] M. George and R. King, "A Robust Speaker Verification Biometric," *Proc. IEEE 29th Ann. Int'l Carnahan Conf. Security Technology*, pp. 41-46, 1995.
[13] S. Haider, A. Abbas, and A. Zaidi, "A Multi-Technique Approach for User Identification through Keystroke Dynamics," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, vol. 2, pp. 1336-1341, 2000.
[14] Z. Changshui and S. Yanhua, "AR Model for Keystroker Dynamics," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, vol. 4, pp. 2887-2890, 2000.
[15] J.A. Robinson, "Computer User Verification Using Login String Keystroke Dynamics," *IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 28, no. 2, pp. 236-241, Mar. 1998.
[16] S.A. Bleha and M.S. Obaidat, "Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users," *IEEE Trans.Systems, Man, and Cybernetics*, vol. 21, no. 2, pp. 452-456, Mar./Apr. 1991.
[17] J. Brault and R. Plamondon, "Segmenting Handwritten Signatures at Their Perceptually Important Points," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 953-957, Sept. 1993.
[18] L. Lee, "Neural Approaches for Human Signature Verification," *Proc. Third Int'l Conf. Document Analysis and Recognition*, vol. 2 pp. 1055-1058, 1995.

[19] R. Martens and L. Claesen, "Automatic On-Line Signature Verification: Discrimination Emphasised," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 657-660, 1997.
[20] M. Mingming and W. Wijesoma, "On-Line Signature Verification Based on Multiple Models," *Proc. IEEE/IAFE/INFORMS Conf. Computational Intelligence for Financial Eng.,* pp. 30-33, Mar. 2000.
[21] T. Wessels and C. Omlin, "A Hybrid System for Signature Verification," *Proc. South African Telecommunications Networks and Applications Conf.,* pp. 5509-5514, 2000.
[22] Y. Xuhua et al., "A Study on Signature Verification Using a New Approach to Genetic Based Machine Learning," *Proc. IEEE Int'l Conf. Intelligent Systems for the 21st Century,* vol. 5, pp. 4383-4386, Oct. 1995.
[23] K. Yue and W. Wijesome, "Improved Segmentation and Segment Association for On-Line Signature Verification," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* vol. 4, pp. 2752-2756, Oct. 2000.
[24] E. Ozcan and C. Mohan, "Shape Recognition Using Genetic Algorithms," *Proc. IEEE Int'l Conf. Evolutionary Computation,* pp. 414-420, May 1996.
[25] E. Ozcan and C. Mohan, "Steady State Memetic Algorithm for Partial Shape Matching," *Proc. IEEE Seventh Ann. Conf. Evolutionary Programming,* pp. 527-536, Mar. 1998.
[26] D. Goldberg, *Genetic Algorithms in Search: Optimisation and Machine Learning.* Reading, Mass.: Addison Wesley, 1989.
[27] C.M. Bishop, *Neural Networks for Pattern Recognition,* New York: Oxford Univ. Press, pp. 332-340, 1995.
[28] A. Zwieseleet, A. Munde, C. Busch, and H. Daum, "Comparative Study of Biometric Identification Systems," *Proc. 34th Ann. IEEE Int'l Carnahan Conf. Security Technology,* pp. 60-63, 2000.