2009 Sixth International Conference on Information Technology: New Generations

Online Payments Using Handwritten Signature Verification

Jarrod Trevathan, Alan McCabe and Wayne Read
Discipline of Information Technology
James Cook University, Australia
Email: jarrod.trevathan@jcu.edu.au, alan@mymait.com, wayne.read@jcu.edu.au

***Abstract*— Making payments online is inherently insecure, especially those involving credit cards where a handwritten signature is normally required to be authenticated. This paper describes a system for enhancing the security of online payments using automated handwritten signature verification. Our system combines complementary statistical models to analyse both the static features of a signature (e.g., shape, slant, size), and its dynamic features (e.g., velocity, pen-tip pressure, timing) to form a judgment about the signer's identity. This approach's novelty lies in combining output from existing Neural Network and Hidden Markov Model based signature verification systems to improve the robustness of any specific approach used alone. The system can be used to authenticate signatures for online credit card payments using an existing model for remote authentication. The system performs reasonably well and achieves an overall error rate of 2.1% in the best case.**

I. Introduction

Credit card fraud and online payment scams are among the leading types of crimes committed in the information age. Typically all that is required is for a dubious individual to be in possession of a stolen card number, and s/he is free to use a card as if it was his/her own. Despite so-called enhanced security measures such as a three-digit security number and the card owner's signature on the back of the card, these mechanisms can be easily overcome. A three-digit security code is only as safe as the original card number and handwritten signatures can easily be forged. Furthermore, verification of the signature is often left to an untrained individual (i.e., a person behind a sales counter), or in the case of online transactions, it is not even used at all.

Handwritten Signature Verification (HSV) is a computerised method for verifying a person's identity by examining the characteristics of, and the manner in which s/he signs his/her signature (see [11]). HSV is considered more natural and less intrusive than other forms of biometric verification such as finger printing and retinal scanning. Static HSV systems observe the general form of a signature looking at characteristics such as shape, size and slant. Dynamic HSV systems measure features specific to the way the person signs his/her name in real time including pen-tip pressure, velocity, duration, etc. The main goals for a HSV system are to reduce the False Acceptance Rate (FAR) – the number of forgeries accepted as genuine; and also to reduce the False Rejection Rate (FRR) – the number of genuine signatures that are rejected as forgeries. The Overall Error Rate (OER) = FAR + FRR.

Statistical models such as the Neural Network (NN) and Hidden Markov Model (HMM) naturally lend themselves to be used as classifiers for HSV systems. Previous papers have dealt specifically with these two complementary approaches to HSV (see [5], [6]). For example, the NN approach from McCabe *et al* [5] is based largely on the analysis of global features of the handwriting, whereas the Hidden Markov Model (HMM) approach from McCabe and Trevathan [6] examines the local aspects. However, there has been limited research into the process of combining, or "fusing", the two methods in order to improve the robustness and performance into a single, powerful HSV system.

Fusion of multiple classifiers is a sub-field of biometrics research that has recently gained in popularity. There are a number of advantages to be gained by combining the output of multiple biometric systems:

- *Improved performance*: If the underlying systems are complementary and the fusion is done well, the performance of the resulting system will be better than that of the constituent systems. This is analogous to consulting a group of experts and making a decision based on multiple opinions, rather than consulting just one expert, and is the primary advantage of combining classifiers;
- *Increased universality*: The resulting system is generally applicable in more situations. This comes about because if one classifier is confused, a decision may still be made using the other(s);
- *Compromises*: Use of multiple systems introduces the possibility of "compromises" if the classifiers disagree. For example, offering restricted access if one of the classifiers verifies a signature and the remainder do not.

The problem of fusing the output of multiple verifiers is not a simple one and is subject largely to the properties of the constituent systems. It is not valid to assume that the combination will always improve performance, for example it is known that a strong system is better used alone than in combination with a weak one [1]. The underlying classifiers should be complementary and redundancy between classifiers may actually degrade accuracy [8].

This paper describes a system for conducting online credit card transactions whereby the authentication of an individual's identity is ascertained using HSV. The system uses complementary statistical models to analyse both the static

978-0-7695-3596-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ITNG.2009.105

and dynamic features of a signature to form a judgment about the signer's identity. This approach's novelty lies in combining output from existing NN and HMM based signature verification systems [5], [6] to improve the robustness of any specific approach used alone. The system performs reasonably well and achieves a 2.1% OER in the best case.

This paper is organised as follows: Section II details how HSV is used to authenticate signatures. Section III discusses the online HSV model for credit card transactions. Section IV presents the methodology and the experimentation performed by combining the models described in [5], [6]. Section V provides some concluding remarks.

II. The HSV Process

Irrespective of the final approach used to perform the verification, most HSV techniques follow five basic phases: *data acquisition*, *pre-processing*, *feature extraction*, *comparison* and *performance evaluation*.

During the first three phases of the signature verification process, most systems generate a reference (or a set of references) for each individual. The reference can be in many forms, including a mathematical model (for example, a NN or HMM), an array of feature values or simply copies of the signatures themselves. The process of creating a reference normally requires several samples of the user's signature to be captured at enrolment or registration time (these signatures are called *sample* signatures or more commonly *reference* signatures) and processed. When a user claims to be a particular individual, they must provide a signature (labelled as the *test* signature) to be compared with the reference for that individual. The test signature undergoes any necessary preprocessing such as noise removal etc. prior to extraction of any features used. The test signature (or, more often, the extracted features) is then compared to the reference and a value is obtained indicating the likelihood that the test signature was produced by the same person that provided the reference signatures. If the similarity is above a predefined threshold value the signer is authenticated, otherwise s/he is rejected as an attempted forger.

A performance evaluation of developed techniques is obviously of great importance and researchers normally use a set of genuine signatures and forgery attempts either collected by them or by someone else, and determine the FRR and the FAR for the technique given the signature database. Obtaining practical estimates of FAR is very difficult since actual forgeries are almost impossible to obtain. Performance evaluations therefore rely on two types of forged signatures. A forgery may be *skilled* if it is produced by a person who has had access to one or more genuine signatures for viewing and/or practice. A forgery is called *zero-effort* or *random* when either another person's genuine signature is used as a forgery or the forger has no access to the genuine signature and is either only given the name of the person whose signature is to be forged or just asked to sign any signature without even knowing the name. Tests on zero-effort forgeries generally lead to much smaller FAR than on skilled forgeries and are really a measure of how easily the system is confused, rather than of forgery defeating ability. In addition to FAR and FRR (and zero-effort FAR), many authors quote an *Overall Error Rate* (OER) or an *Equal Error Rate* (EER). The OER is simply the sum of FAR and FRR whereas the EER comes about because of the inverse relationship between FAR and FRR where one of these rates decreases as the other increases (and vice versa). The EER is the point at which these two error rates cross.

The value of performance evaluation in assessing a HSV system is obvious. However, it is not always a true indicator of the performance of the technique for a variety of reasons. The primary problem with testing procedures is that often the test signatures do not adequately represent the population at large. There is a great deal of signature variability according to the author's nationality, age, time, habits, psychological or mental state, and physical and practical situations. Building a test database of signatures that is representative of real-world applications is quite a difficult task given the effort involved to find a large number of people that will willingly provide ten or twenty samples of their signature in several different sessions. Most people are reluctant to have their signatures stored in a computer or given to others to practice forging them. Additionally, few people can make themselves available to provide the signatures over many different days (as is required to capture the day-to-day variability of signatures).

Most test databases are built using signatures from volunteers from the research laboratory where the HSV research has been carried out and as a result have very few signatures from people that are old, disabled, suffering from a common disease (for example arthritis), or poor. Percentages of such people in the population are significant and these are the people whose signatures are likely to pose the greatest challenge for a HSV technique. The inconsistent signing environment in the real world poses further challenges for HSV as test databases are often created in very controlled situations and settings - something that is not the case in the real world.

The lack of common test databases also prevents any meaningful comparisons between different HSV systems. Most researchers have their own test signature databases with a varying number of genuine signatures. Some have skilled forgeries while others do not, some have screened the signature database and removed several signatures that for some reason were not acceptable while others have done no screening. The number of signatures used in building a reference signature often varies. Different tests and thresholds have been used and even different definitions of FAR and FRR have been used. S ome studies use a different threshold for each individual while others use the same threshold for the entire database. This is a rather sad state of the art in HSV.

After a set of features has been selected and extracted, there is no real need to store the reference signature and only the feature values of the reference signature need to be stored. Also, when a test signature is presented, only the feature values are needed, not the signature itself. This often saves on storage (which may be at premium if, for example, the reference signature needs to be stored on a card) and is the reason that

signature representation using features is sometimes referred to as *compression* of the signature. In the methods that use feature-based comparison, selection of features and extracting the best subset once a set of features has been identified are major research tasks. Some of the problems related to these tasks are:

- How many features are sufficient?
- Do the features require some transformation to be applied to the signature data, for example resizing, deslanting, time-warping?
- How many sample signatures will be used in computing the reference?
- Would the reference signature be updated regularly? What would be the procedure for this?
- How is the distance between a test signature and the reference going to be computed?

Features, once decided upon and extracted from the test signature, then must be compared to the reference in some meaningful way. The focus of following discussion is on this comparison process.

Assuming that the reference is based on a set of sample signatures, and for each element of the set of selected features the mean and standard deviation of the feature values have been computed. The reference therefore consists of two vectors: a vector (R) of the means of the feature values and a vector (S) of the standard deviations. Clearly, to obtain accurate estimates of the mean and the standard deviations it is necessary to use several, perhaps a minimum of five, sample signatures. A larger set of sample signatures is likely to lead to a better reference and therefore better results but it is recognised that this is not always possible.

The issue of how many features a particular method needs to use to obtain a reliable verification result is a difficult one. The natural temptation is to include more and more features in the hope of improving performance and, as was noted earlier, some researches have proposed more than 90 features. Depending on the nature of the method being used, it is highly possible that the use of *too many* features may actually degrade the performance of a system. For example, at a basic level, the storage needed to record the reference details may exceed the available capacity (if the storage medium is some kind of "smart card" or any other device with limited memory). A large number of features also increases the complexity of many aspects of the problem (learning, comparison etc.) and will result in slower system execution. Additionally, in large feature sets where choosing of features has been less selective, it is likely that several of the features will be less representative of the natural properties of the signature. The effect of this is that the non-representative features will cause genuine signatures to appear further away from the reference than they should be and may well even cause forgeries to be closer to the reference than they should be (i.e., "noise" is being introduced into the dataset). Obviously this is going to degrade the system's accuracy.

It should be noted that some techniques like NN learning attempt to automatically associate a weighting or importance with the different features. The result is that, from a large feature set, those which are less important are given a lower weighting and thus and have less effect on the overall classification of that signature. Use of many features therefore will not of itself degrade the system, but if less relevant features are allowed to influence the classification process then a loss of performance will result.

As for the number of sample signatures required to build a reference, this remains an open question. There is a strong relationship between the quality of the reference and the number of sample signatures used to create this reference. Obviously, all other things being equal, a reference built using many sample signatures is going to be more representative of the signer's natural variation than one built using few. Unfortunately, requiring that a user provide several (e.g., more than ten) signatures would be of great annoyance to that user (in addition to larger computational and storage requirements). There is a degree of tradeoff between the level of reference quality and the level of user annoyance.

A further issue with regard to sample signatures is that of updating. Any real-world HSV system needs to adapt to the changing signature of a user. Although most users will not undergo a notable change in their signature once the natural style has been established. A practical HSV system should make some attempt to update references for users in the database. The process of signature evolution is difficult to simulate in laboratory conditions and as a result systematic studies of adaptive systems are almost impossible. Techniques to handle the evolution include the re-taking of sample signatures at some regular interval (which is inconvenient for users) or incorporation of verified signatures into the reference set (at the risk of incorporating forgeries).

III. Online Payment Model

This section presents the system for performing remote authentication via HSV. It outlines the basic remote HSV model, network-related security issues, and finally the credit card HSV model itself.

A. The Basic Remote HSV Model

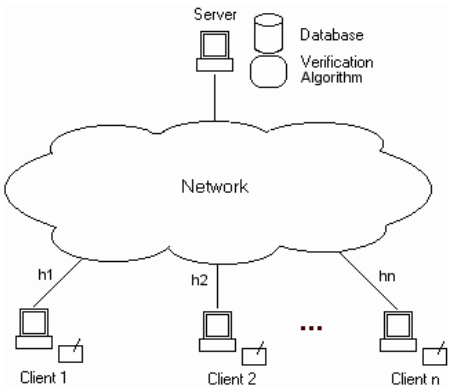

Fig. 1. *Remote Handwritten Signature Verification*

Figure 1 presents a client/server model for performing remote HSV (refer to Trevathan and McCabe [12] for more details). The main components of the system include:

- A set of clients $C = \{c_1, c_2, ..., c_n\}$ who each possess a unique handwritten signature h; h_i denotes the signature corresponding to c_i, $1 \leq i \leq n$;
- A *trusted* server S to which clients can make requests for verification. The server contains a verification algorithm V, and secure reference database;
- A network providing a bi-directional communication link between each client and the server.

A client, c_i, provides a signature for verification (in which case he/she is trying to establish or "prove" his/her identity). Alternately, c_i can be the recipient of the verification results (in which case the server sends them the verified identity of the signer).

The server, S, verifies signatures presented on the client's behalf. The verification algorithm can be updated on the server and hence technology may be improved without requiring any changes to the client software. Furthermore, having the verification software on the server removes the bias that would exist if a client also contained the same software. For example, there is a possibility that malicious users may tamper with or simulate the verification algorithm.

The database is used to store handwritten signature reference files, client information and security logs.

The verification algorithm, V, makes a binary decision to ascertain whether the client is genuine. The algorithm must be able to successfully authenticate legitimate signatures and identify forgeries. An outline of the verification algorithm is presented in Section IV.

The network links clients to the server. This network is assumed to be insecure and any information passing through it can be tampered with.

B. Network Security Issues

In a network environment the signature data is at risk from eavesdropping, interception, modification, fabrication and disruption. Measures such as encryption, *digital* signatures and time stamping, are needed to provide protection against these threats. This section examines network security requirements for the HSV scheme presented in this paper.

A client is provided with an initial session key, k_i, during registration. This allows encrypted communication with the server using any off-the-shelf symmetric cryptosystem. Key updates are performed using Diffie-Hellman key exchange [2].

The results of verification are signed using a digital signature (not to be confused with handwritten or electronic signatures). S publishes a public key S_{ku}. This enables S to sign the results of verification using the private key S_{kr}. Anybody can verify the signature using S_{ku}. This can be achieved using RSA [10] or ElGamal [3] signatures.

Time stamping is required to prevent an interceptor from resending captured messages. A time stamp is appended to the handwritten signature data before encryption. When a message is received, its time stamp is examined to determine how old the message is and if it has been previously received. This presents a unique problem to HSV as any captured electronic signature could be reused indefinitely. A remedy outlined in McCabe and Trevathan [7], is to use a signed password (rather than a signature) to verify a user. This enables compromised handwritten passwords to be changed, rather than handwritten signatures.

S logs all information regarding verification. Users can access these logs in accordance with S's policies.

C. Credit Card and Online Payments

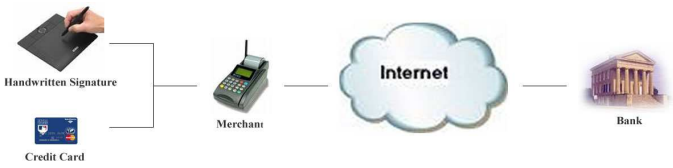

Fig. 2. *Online Payment Model.*

Figure 2 illustrates how the system is used for making payments using a credit card. The user enters all relevent card details (i.e., card number, expiry date, name on the card) and then provides a sample of his/her handwritten signature. This information is transmited (via a POS device or web page) by the Merchant over the web. The signature is then verified by the bank or a trusted server that hosts the verification algorithm. Payment goes ahead if the signature is verified as being legitimate. Notification is then transmitted back to the Merchant via the web. As an additional safeguard, the Merchant can still check the user's authenticity via the traditional offline methods as well.

IV. The HSV Algorithm and Experimental Results

This section presents the methodology and the experimentation performed in the combination of the models described in [5], [6]. The combination of the two classifiers essentially involves the development of a whole new classifier that has two input values: the confidence value output by the NN system and the normalized log-likelihood output by the HMM system. Figure 3 illustrates the basic approach.

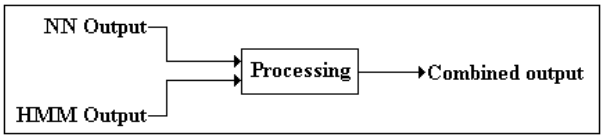

Fig. 3. *The combination of models described in previous papers.*

A. Experimental Setup

The signature database used in experimentation is described in [5]. The type of the two input values differs in that the value provided from the NN is a confidence measure and the HMM value is a probability. However, the domains of the

Acceptance Mechanism	FAR	FRR	OER
Unanimous	0.9%	2.1%	3.0%
Disputed	2.2%	1.6%	3.8%

TABLE I

The results using the two different voting mechanisms to combine the classifiers.

input values are similar in that they are both in the zero-to-one range with the same interpretation: a low score implies a low degree of confidence that a given signature is genuine, while a high score implies a high degree of confidence. What follows is a description of the different methods of combining the two models, grouped according to category.

1) Voting Schemes: These are the simplest methods of combining the HMM and NN output and involve the combination being performed at the decision level. The only input taken in from the two models is the binary decision rather than the confidence values. With two different classifiers as input there are two verification scenarios than can be employed:

- *Unanimous acceptance*: This means that the combined classifier should accept a test signature only if both constituent classifiers accept it. Put another way, the test signature should be rejected if *either* system classifies it as non-genuine. The ideal effect of this is that fewer forgeries will be accepted (as it is less likely that they will deceive both classifiers) without rejecting many more genuine signatures (as well-performed genuine signatures should be accepted by both classifiers). Table I presents the results obtained using this approach, which include the FAR, FRR and OER.
- *Disputed acceptance*: This setup results in a signature being accepted if it is accepted by *either* classifier (or both). Ideally, forgeries are still rejected (as both of the constituent classifiers are quite adept at detecting forgeries) while slightly lower quality signatures from genuine users are still accepted (as they may still contain enough characteristic information to be accepted by at least one of the classifiers). The results obtained using this approach are also presented in Table I. Unanimous rejection by both classifiers should obviously result in the signature being rejected.

When discussing the results presented in Table I it is useful to recall that the best OERs for our NN [5] and HMM [6] systems are 3.3% and 3.5% respectively. Relative to the individual system error rates the "Unanimous Acceptance" results were an improvement over the most successful individual system, whereas the "Disputed Acceptance" approach actually degraded the overall performance. As expected, the unanimous approach resulted in a much lower FAR as it was far more difficult for a forgery to deceive both classifiers. Fortunately, there was not a greatly adverse affect on the FRR, which meant the OER was improved. The disputed approach results in the FRR slightly improving, however, the FAR increases substantially and the OER suffers as a result. Inspection of the rejected genuine signatures offers an explanation for the lack of significant improvement in the FRR in that these signatures are either poorly written or differed greatly from the signatures given as a reference for that user. As a result they tend to be rejected by both classifiers.

However, the results above do not give a definitive answer as to which acceptance mechanism is most suited to combining HSV systems. The approach to use depends largely on the environment in which the signatures will typically be provided. If the environment is casual as in a general purpose system, it is likely that the test signatures will be of slightly lower quality and the disputed approach is more forgiving and more appropriate. If the environment is a formal, high security one, then the added security of the unanimous approach is likely to result in more desirable performance. It is believed that the formal environment in which the handwritten signature database was captured contributed to the unanimous approaches superior performance.

A voting mechanism may also be used in granting different levels of access depending on the level of signature acceptance. That is, if both classifiers reject the test signature then no access is granted, if both accept the test signature then full access is granted or if the classifiers are in disagreement then partial or restricted access is granted.

2) Confidence-based Approaches: This section discusses the various techniques used to combine the confidence outputs from each of the classifiers. All explored methods and resulting error rates are presented below.

– **Weighted sum**: The weighted sum rule (sometimes referred to as simply the sum rule) involves taking the weighted sum of the individual scores from the classifiers to achieve the overall score for the combined system:

$$S_{combined} = W_{NN} \cdot S_{NN} + W_{HMM} \cdot S_{HMM}$$

where $S_{combined}$ is the final score for the combined model, W_{NN} and S_{NN} are the weight and score for the NN model and W_{HMM} and S_{HMM} are the weight and score for the HMM. The score values from each of the models are obtained independently and are a measure of the confidence that each model has in the test signature being genuine.

The weight and threshold values are the same for every user and are obtained in a joint training phase. This phase involves an exhaustive search that tests all weight values in the range [0,1] (with increments of 0.01) for each classifier, with the constraint that the sum of the weights is always 1. The combined scores are obtained for each weight combination and compared to a threshold. The test signature is accepted if the score is above this threshold and rejected otherwise. The threshold value and weight pairing that gave the lowest global (i.e., over all users in the database) OER are used for all further experiments. These values are 0.42 for the threshold and 0.62 and 0.38 for the HMM and NN weights respectively.

This approach worked quite well and improved the OER to 2.7% (1.5% FAR and 1.2% FRR).

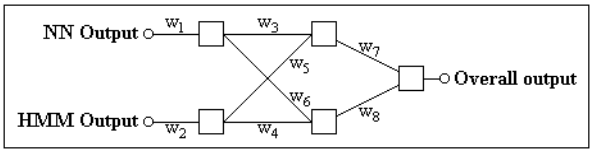

Fig. 4. *The MLP structure that produced the lowest OER when combining the constituent systems. Each of the weight values W_i is optimised via a learning algorithm.*

Combination Method	FAR	FRR	OER
Weighted sum	1.5%	1.2%	2.7%
Product rule	1.3%	1.5%	2.8%
Mean transformation	1.8%	2.1%	3.9%
Decision trees	5.5%	2.9%	8.4%
Multi-layer perceptrons	1.3%	1.7%	3.0%
User-specific weighted sum	1.1%	1.0%	2.1%

TABLE II

The resulting error rates using the different confidence-based approaches to combining the classifiers.

– **Product rule**: This is quite similar to the sum rule and involves taking the weighted product of classifier scores:

$$S_{combined} = W_{NN}.S_{NN} \times W_{HMM}.S_{HMM}$$

Weight values and thresholds are obtained in the same way as they were for the sum rule approach. The product rule represents the joint probability distribution of the values extracted by the classifiers. This approach produces very similar results to that of the sum rule and obtained a 2.8% OER (1.3% FAR and 1.5% FRR).

– **Mean transformation**: This is essentially a special case of the weighted sum with the weights set to 0.5 and is used in [4] to combine classifiers for fingerprint verification. This transformation simply takes the mean of the two classifier scores for a test signature:

$$S_{combined} = \frac{S_{NN} + S_{HMM}}{2}$$

The mean approach did not perform well in this instance, returning an overall error rate of 3.9% (1.8% FAR and 2.1% FRR), which is worse than both individual classifiers.

– **Decision trees**: Results were obtained using the *C*5.0 program [9] to generate a decision tree from a training set of classifier score pairs. Five genuine score pairs were used as a reference and the score pairs from genuine signatures of twenty-five other users in the database were used as negative examples (the twenty-five users were selected in a similar way to training forgeries selected in [5]). The data seemed to be insufficient for accurate construction of decision trees and *C*5.0 did not perform well using only five genuine references, resulting in a 8.4% OER (5.5% FAR and 2.9% FRR).

– **Multi-layer perceptrons**: Multi-layer Perceptrons (MLPs) are used in [5] to build one of the constituent HSV systems. It was theorised that a non-linear classifier such as a MLP may be able to achieve better classification by capturing a more insightful relationship between the two confidence measures. The basic model structure consists of a three-layer network with two input units and one output unit. The back-propagation algorithm was used to train the model and experimentation was done with different numbers of nodes in the hidden layer, but the most successful structure found contained two hidden nodes (Figure 4 illustrates the structure).

The makeup of the training set (in terms of positive and negative examples) was identical to that used for the decision tree approach described previously. The MLP's performance over the training set was very good but did not generalise as well to the test set, most probably because of the small amount of input data. The OER using this approach was 3.0% (1.3% FAR and 1.7% FRR).

– **User-specific weighted sum**: This is the most successful of the approaches to combining model scores. The method here is similar to that used in the "weighted sum" approach, modified to apply to individual users. The basic algorithm for adjusting the user-specific weights is as follows:

1) For user i, vary weights $W_{NN,i}$ and $W_{HMM,i}$ over the range [0,1] (with increments of 0.01) with the constraint that $W_{NN,i} + W_{HMM,i}$ equals 1;
2) The overall score used for verification is then:

$$S_i = W_{NN,i} \times S_{NN,i} + W_{HMM,i} \times S_{HMM,i}$$

3) S_i is compared to a user-specific threshold T_i for each user and the test signatures is accepted if $S_i > T_i$ and rejected otherwise;
4) Choose the set of weights and thresholds that minimises the *total error rate* associated with the overall scores.

Here $W_{NN,i}$ refers to the user-specific weight associated with the NN output for user i and $W_{HMM,i}$ refers to the user-specific weight associated with the HMM output for user i. Similarly $S_{NN,i}$ refers to the score (i.e., output or confidence) from user i's NN and $S_{HMM,i}$ the score from user i's HMM. S_i refers to the overall score value for user i. The *total error rate* referred to in step 3 of the algorithm (not to be confused with the *overall* error rate) is the sum of the individual error rates calculated during the training phase. Details of the weight and threshold selection appear below.

The overall score S_i is obtained for each weight combination in the range [0,1] (with increments of 0.01), with the constraint that the sum of the weights is always 1. In an extended training phase, error rates are calculated using the original reference signatures as genuine attempts (i.e., the training and testing databases remain separate) and a set of thirty-five forgeries (obtained from the other users in the database in the same manner as described in [5]). These error rates are calculated by exhaustive experimentation with

Model	FAR	FRR	OER
Neural network	1.1%	2.2%	3.3%
Hidden Markov model	1.2%	2.3%	3.5%
Weighted sum	1.5%	1.2%	2.7%
User-specific weighted sum	1.1%	1.0%	2.1%

TABLE III

The most successful results for each of the different model scenarios used during development.

threshold values, varying the threshold in the range [0,1] (with increments of 0.01) for each weight pairing. The threshold and weight values triple that produces the lowest OER is then fixed for that user. Often a range of weight values results in an OER of zero for a particular user - in this case the median of each weight range is used. Similarly, the median threshold is used when there is a range of threshold values resulting in the equal lowest error rate.

OERs for each user are then calculated using the fixed weight and threshold values with the previously unseen genuine signatures and skilled forgeries being used as test signatures. This approach improved the OER to 2.1% (1.1% FAR and 1.0% FRR) and returned an EER of 1.1%. All error rates quoted in following sections will be based on this user-specific weighted sum approach unless otherwise specified.

Table II summarises the results of all of the confidence-based methods of combination.

A summary of all of the developmental models appears in Table III, including the NN alone, the HMM alone, both models combined via the weighted sum rule and both models combined via the user-specific weighted sum approach. As can be seen, the combination of models resulted in an increased performance over both individual models, with the user-specific weighted sum approach being most successful.

V. Conclusions

This paper describes a system for conducting online credit card transactions whereby the authentication of an individual's identity is ascertained using HSV. Once registered, a user presents a test signature to a merchant at the point of sale. The signature is then securely sent across a network to a server where the verification algorithm resides. The server grants the merchant payment if the signature is genuine, otherwise it rejects the proposed transaction. Using HSV for online payments greatly enhances the security of credit card transactions compared to the existing flawed process whereby an untrained human manaually compares the test signature to the signature on the back of the credit card.

The HSV algorithm is based on complementary statistical models. The system analyses both the static features of a signature (e.g., shape, slant, size), and its dynamic features (e.g., velocity, pen-tip pressure, timing) to form a judgment about the signer's identity. This approach's novelty lies in combining output from existing Neural Network and Hidden Markov Model based signature verification systems to improve the robustness of any specific approach used alone. The system performs reasonably well and achieves a 2.1% OER in the best case.

Future work involves the analysis of the independence of the two classifiers used in this study (and indeed the fusion classifier in general). The methodologies described in our previous papers [5], [6] present demonstrably different classifiers, both in terms of input features and learning/classification techniques. Further work would include a detailed investigation as to the style of signatures that are correctly classified by both individual systems, and the tradeoff between individual classifier accuracy and independence of classifiers. In addition, there are more practical implmentation issues pertaining to network security that must be investigated. This includes timestamping test signatures to prevent their reuse, and/or using a handwritten password rather than a signature to enhance security.

References

[1] J. Daugman. Biometric Decision Landscapes. *Technical Report No. TR482*, University of Cambridge Computer Laboratory, 2000.

[2] W. Diffie and M.E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22, pp. 644-654, 1976.

[3] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31, pp. 469-472, 1985.

[4] G. Marcialis, F. Roli and P. Loddo. Fusion of Multiple Matchers for Fingerprint Verification. *Proceedings of the Workshop on Machine Vision and Perception*, Italy, 2002.

[5] A. McCabe, J. Trevathan and W. Read. Neural Network-based Handwritten Signature Verification. *Journal of Computers*, Vol. 8, No. 3, pp. 9-22, 2008.

[6] A. McCabe and J. Trevathan. Markov Model-based Handwritten Signature Verification. *Proceedings of the IEEE/IFIP International Symposium on Trust, Security and Privacy for Pervasive Applications*, pp. 173-179, 2008.

[7] A. McCabe, J. Trevathan and W. Read. Handwritten Signature Verification using Complementary Statisitical Models. *Journal of Computers*, (to appear) 2009.

[8] S. Prabhakar and A.K. Jain. Decision-Level Fusion in Fingerprint Verification. *Pattern Recognition*, Vol. 35, No. 4, pp 861-874, 2002.

[9] R. Quinlan. *Data Mining Tools See5 and C5.0.* http://www.rulequest.com/see5-info.html, 2002.

[10] Rivest, R., Shamir, A. and Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21 (2), pp. 120-126, 1978.

[11] H.E.S. Said, K.D. Baker and T.N. Tan. Personal Identification Based on Handwriting. *Proceedings of the 14th International Conference on Pattern Recognition*, Brisbane, Australia, pp. 1761-1764, 1998.

[12] J. Trevathan and A. McCabe. Remote Handwritten Signature Authentication, *In Proceedings of the* 2^{nd} *International Conference on e-Businees and Telecommunications Engineers*, pp. 335–339, 2005.