AUTHENTICATING COMPUTER ACCESS BASED ON KEYSTROKE DYNAMICS USING A PROBABILISTIC NEURAL NETWORK

Kenneth Revett[1], Florin Gorunescu[2], Marina Gorunescu[2], Marius Ene[2] Sérgio Tenreiro de Magalhães[3], and Henrique M. D. Santos[3]

[1]*Harrow School of Computer Science, University of Westminster, London, UK*
revettk@westminster.ac.uk

[3] *Universidade do Minho Department of Information SystemsCampus de Azurem*
4800-058 Guimaraes, Portugal
{psmagalhaes, hsantos} @dsi.uminho.pt

[2]*Department of Mathematics, Biostatistics and Computer Science, University of Medicin and Pharmacy of Craiova, Romania*
{fgorun, mgorun, enem}@umfcv.ro

Abstract: Most computer systems are secured using a login id and password. When computers are connected to the internet, they become more vulnerable as more machines are available to attack them. In this paper, we present a novel method for protecting/enhancing login protection that can reduce the potential threat of internet connected computers. Our method is based on and enhancement to login id/password based on keystroke dynamics. We employ a novel authentication algorithm based on a probabilistic neural network. Our results indicate that we can achieve an equal error rate of less than 5%, comparable to what is achieved with hardware based solutions such as fingerprint scanners and facial recognition systems.

1. Introduction

The traditional method for authentication in most information systems that are computer based is the login/password (e.g. class C security minimum). There are countless reports on how such class C protected devices have been breached, resulting in financial losses and a dimunition in the faith people have when transacting business over the internet (Peacock, 2004). Researchers and industry alike have sought ways to enhance the security of computers that currently house major financial and scientific data throughout the world. A new industry has been created – with the sole purpose of providing enhanced protection from piracy – the biometrics industry. Currently, there are two major forms of biometrics: those based on physiological attributes such as fingerprints, iris, and retinal scanners and behavioural biometrics: based on voice recognition, signature verification and keystroke dynamics (see Jain, 2001, 2003 for a nice review).

Physiological biometrics is based on the notion of what we are – we each possess unique fingerprints (even identical twins differ in their fingerprint patters) and are therefore thought to be spoof proof. But current literature reports indicate that fingerprints can be spoofed (Jain, 2003, Peacock, 2004). Even though fingerprint scanners are becoming more reliable and cheaper to acquire, they still are subject to noise and wear and tear. They require replacement approximately once a year and are difficult to place on remote access systems, such as a home computer used in a credit card purchase over the internet. Iris scanners and more noise tolerant, but are certainly more expensive than fingerprint scanners. In addition, they are (or at least appear to be) more intrusive – a very important factor in a biometric. Any biometric solution that is to be used on the internet (and therefore accessed by potentially 100s of millions of users) must be effective and yet very unobtrusive. In the ideal case, we wish to provide a secure site that provides no hint that it is being heavily

protected. Alerting users to the fact that sites on the internet are subject to attack instills suspicion – the exact opposite sentiment the originators had for the internet. It was designed to provide a medium where scientific information could be exchanged in an environment that was purely based on academic freedom. In today's world, the ideals can still exist, but may require a medium that is fortified with enhanced security features – the ethos of biometrics.

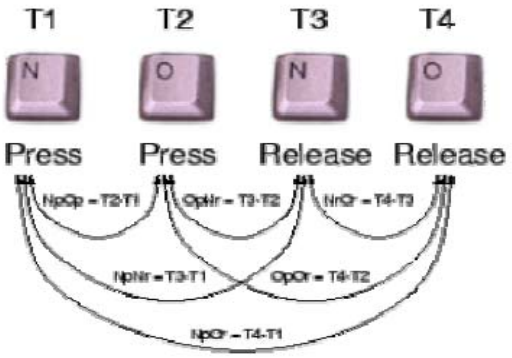
In this paper, we present evidence that keystroke dynamics is a viable biometric – that provides security on par with physiological methodologies. In addition, it provides enhanced security in a very low-profile manner that is acceptable by the majority of users – virtually everyone is used to entering authentication details such as a login id and password. The remaining question is whether keystroke dynamics – the typing style – measured in terms of keystroke/keypress duration and keyboard latency, combined with state-of-the-art machine learning techniques are sufficiently robust to provide the ability to discriminate between an authentic and imposter. If evidence can be shown that this is indeed the case – then behavioural biometrics – specifically keystrokes dynamics may provide a unique and effective solution to user authentication that can be used on personal machines as well as internet based applications.

In this paper, we will describe the basics of keystroke dynamics, followed by a machine learning algorithm – a probabilistic neural network on a dataset consisting of authentic and imposter login attempts. We then present some key results of our work and a brief discussion and conclusion section.

2. Keystroke dynamics

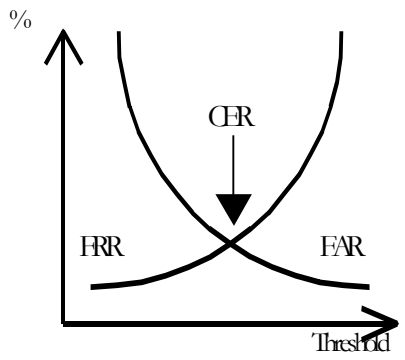
Keystroke dynamics is a class of behavioural biometrics that captures the typing style of a user. By typing style, we mean it examines how long it takes to type the login id/password, how long we depress a key, and how long we take to type successive keys – this is called a digraph. Figure 1 illustrates the classic example of the

Figure 1. The concept of a digraph – and the various combinations that can be extracted and used for biometric authentication. In this aprticular example, the digraph is based on the character sequence 'no.'



data that can be extracted by entering two keys on a standard keyboard. By collecting all possible digraphs from the login id/password – one can develop a model of how the person types these credentials. Usually, there is an enrollment exercise, where the user is asked to enter his/her login id/password until a steady value for each digraph is obtained. Once this data has been collected, a reference 'signature' is obtained for this user. The reference is then used on subsequent login attempts – a user with that particular login id/password combination has their keystroke dynamics extracted and then compared with a stored reference value. If they are within a prescribed tolerance limit – the user is authenticated. If not – then the system can decide whether to lock up the workstation – or take some other suitable action. When devising such a biometric solution – there is always a trade off between being overly stringent – rejecting every attempt to login in and being overly lenient – allowing imposters to access the computer. The former is usually reported as a measure of false rejection – a type I error and the later a false acceptance or type II error. Another measure – called the cross over error rate (CER) - sometimes referred to as the equal error rate (EER) is also reported – they provide a measure of how sensitive the biometric is at balancing ease of use for the authentic user while at the same time reducing the imposter

Figure 2. The cross-over error rate is indicated as the intersection between the FAR v FRR – when measured against a threshold.



access rate. All extant biometric systems yield a trade-off between these two measures – those that reject imposters effectively (low FAR) are usually accompanied by a high FRR and vice versa. Figure 2 depicts a typical plot of FAR/FRR and indicates the CER point – where the two plots intersect. The critical research issue is 'how can we lower the CER?' In the next section, we describe the historical context in which keystroke dynamics has developed highlighting advances made in the methodological approaches to this critical question.

2.1. Background

In 1980 Gaines (Gaines, 1980) presented a report of his work to study the typing patterns of seven professional typists. The small number of volunteers and the fact that the algorithm is deducted from their data and not tested in other people later, results on a lower confidence on the FAR and FRR values presented. But the method used to establish a pattern was a breakthrough: a study of the time spent to type the same two letters (digraph), when together in the text. Since then, many algorithms based on Algebra and on Probability and Statistics have been presented. Joyce Gupta presented in 1990 (Gupta, 1990) an algorithm to calculate a value that represents the distance between acquired keystroke latency times and correspondent times previously stored. In 1997 Monrose and Rubin use the Euclidean Distance and probabilistic calculations based on the assumption that the latency times for one-digraph exhibits a Normal Distribution (Monrose, 1997). Later, in 2000, they also present an algorithm for identification, based on the similarity models of Bayes, and in 2001 they present an algorithm that uses polynomials and vector spaces to generate complex passwords from a simple one, using the keystroke pattern (Monrose, 2001).

Various fuzzy logic algorithms have been applied – mapping the variability in ones typing patterns to a fuzzy concept. For instance, Hussein et al (Hussien, 1989, de Ru et al., 1997) use a combination of fuzzy clustering algorithms - obtaining an error rate of approximately 5-10% - depending on the number of samples they acquired per login id/password combination. Another study (Tapiador, 1999) employed a fuzzy rule set in order to classify login id/password combinations with somewhat better success than Hussein – although they report only their preliminary results.

Techniques based on neural networks have been explored – focusing on ART-2 and multi-layer perceptrons trained with the backpropagation algorithm. For instance, Obadiat provides data that suggests that the error rate can be reduced to approximately 2.4-4.2%, depending on the exact pre-processing performed using a non-standard neural network (Obadiat, 1997). Sung et al., has also applied neural networks (using standard backpropagation) to keystroke dynamics, generating error rates on the order of 2-4% (Sung, 2006).

Other machine learning approaches, based on support vector machines (SVM) have been used to address the classification problem presented by keystroke dynamics. De Oliveira et al (de Oliveira, 2005) have applied SVM to a small keystroke dataset and compare their results to standard neural network technology. The authors claim that the SVM classifier is more efficient and at least as accurate as neural network technologies. Sung et al. have also

applied SVM to this domain, reporting an error rate of approximately 8-10% (Sung et a., 2006).

Lastly, Revett et al. have used the rough sets induction algorithm to extract rules that form models for predicting the validity of a login id/password attempt (Revett et al., 2005). The results indicate that the error rate can be as low as 2-4% in many cases.

The algorithms cited are a small example of the many approaches used to find adequate keystroke dynamics algorithms with a convenient CER. Many others could also be referred, all with different evaluation methods, different number of users involved (usually a limited number of users), different number of keystrokes required to enroll the system and different number of repetitive operations required to authenticate and/or identify the user. This diversity in the algorithm parameters and in the evaluation method makes the task of comparing their results a very difficult one. Furthermore, there is, in this subject, no concept of what is a representative data sample. The same algorithm presents different results when tested with different volunteer groups. The only way to compare two algorithms is to test it against the same group.

Envisaging wide scale applications, like web-based applications (where this method is not executable now) one must consider the results only if the test user group's size is considerably large. In this application domain one must remember that the computational effort necessary to execute the algorithm is a critical factor. This is one of the driving forces behind the approach we have adopted in this paper – we have implemented a probabilistic neural network (PNN) to classify whether a given login id/password combination belongs to the authentic user or an imposter. A PNN is a highly accurate classifier and is very efficient for small datasets – here the dataset entails the details for an authentic user and a set of imposters selected randomly. In these circumstances, we will provide evidence that this machine learning algorithm is as accurate as the leading results and is more efficient in many ways. We describe the basis of the PNN algorithm in the next section.

2.1. Probabilistic Neural Networks

The PNNs are basically classifiers (Specht, 1988). The general classification problem is to determine the category membership of a multivariate sample data (i.e. a *p*-dimensional random vector **x**) into one of *q* possible groups Ω_i, i = 1, 2,…, q, based on a set of measurements. If we know the probability density functions (p.d.f.) $f_i(\mathbf{x})$, usually the Parzen-Cacoulos or Parzen like p.d.f. classifiers:

$$f_i(x) = \frac{1}{(2\pi)^{p/2}\sigma^p} \cdot \frac{1}{m_i} \cdot \sum_{j=1}^{m_i} \exp\left(-\frac{\left\| x - x_j \right\|^2}{2\sigma^2} \right) \quad (1)$$

the *a priori* probabilities $h_i = P(\Omega_i)$ of occurrence of patterns from categories Ω_i and the *loss* (or *cost*) parameters l_i associated with all incorrect decisions given $\Omega = \Omega_i$, then, according to the Bayesian decision rule, we classify **x** into the category Ω_i if the inequality $l_i\ h_i\ f_i(\mathbf{x}) > l_j\ h_j\ f_j(\mathbf{x})$ holds true. The standard training procedure for PNN requires a single pass over all the training patterns, giving them the advantage of being faster than the feed-forward neural networks (Specht, 1988).

Basically, the architecture of PNN is limited to three layers: the *input/pattern layer*, the *summation layer* and the *output layer*. Each input/pattern node forms a product of the input pattern vector **x** with a weight vector W_i and then perform a nonlinear operation, that is $\exp[-(W_i - x)^\tau (W_i - x)/(2\sigma^2)]$ (assuming that both **x** and W_i are normalized to unit length), before outputting its activation level to the summation node. Each summation node receives the outputs from the input/pattern nodes associated with a given class and simply sums the inputs from the pattern units that correspond to the category from which the training pattern was selected, $\sum_i \exp[-(W_i - x)^\tau (W_i - x)/(2\sigma^2)]$. The output nodes produce binary outputs by using the inequality:

$$\sum_i \exp[-(W_i - x)^\tau (W_i - x)/(2\sigma^2)] > \sum_j \exp[-(W_j - x)^\tau (W_j - x)/(2\sigma^2)] \quad (2)$$

related to two different categories Ω_i and Ω_j.

The key to obtaining a good classification using PNN is to optimally estimate the two parameters of the Bayes decision rule, the misclassification costs and the prior probabilities. In our practical experiment we have estimate them heuristically. Thus, as concerns the costs parameters, we have considered them depending on the average distances D_i, inversely proportional, that is $l_i = 1/D_i$. As concerns the prior probabilities, they measure the membership probability in each group and, thus, we have considered them equal to each group size, that is $h_i = m_i$. As in our previous work, we employed an evolutionary technique based on the genetic algorithm to find the smoothing parameters (see Gorunescu et al, 2005 for implementation details). In the next section, we describe the experimental methods, with a brief description of the dataset.

3. Methods

The dataset we examined consisted of a group of 50 subjects (all university students in a computer science department) – 20 acting as authentic users and the balance (30) acting as imposters. We asked the users to enter a login id/password of their choice (limit 15 characters for each) with an enrollment of 10 trials. We utilised only their digraph latencies and scan code of the characters contained within their login id/passwords. The data samples were collected over a 14-day period, throughout various periods of the day. We maintained a running average of digraph values – where the oldest sample of 10 was replaced – leaving a set of 10 most recent login ids/password attempts. We invited 30 students to act as imposters, requesting that they attempt to hack into someone's account by giving them their account holders login id/passwords. They were given 2 days to attempt over 500 trials of logging into all 20 authentic accounts and the results were recorded. We then used our PNN to generate a classifier that would be able to discriminate between an authentic user and an impostor – using subsets of the data thus obtained. We cross validated our data in that we sampled with replacement until all datapoints were used in the classification and we report the average results from these experiments. The particular version of the PNN we employed in this paper was the same as that employed in previous work (Gorunescu et al., 2005, Revett et al., 2005). We also applied a modified version of our PNN algorithm, that used separate smoothing factors for each class (authentic and imposter). We report both results in this work – and found that using a separate smoothing factor provided consistently better results.

4. Results

We first describe an experiment where we examined which division used in the PNN gave us the best classification accuracy to determine which division provided the best accuracy. We selected random samples for training and testing (70/30 in this case) and applied our PNN algorithm to these random samples. The data in Table 1 indicate that the classification accuracy was essentially independent on the number of divisions employed for this dataset. Please note that the modified PNN algorithm yielded consistently higher results than one that employed the same smoothing factor for both classes – those results are not presented.

Table 1. The classification results from the application of the modified PNN as a function of the number of divisions used. The overall accuracy was 0.099 (9.9%) for this experiment.

Divisions	TRAINING error	TEST error
10	0.0107	0.1153
20	0.0001	0.0884
30	0.0000	0.0942
40	0.0003	0.0923
50	0.0100	0.0961
60	0.0009	0.1153
70	0.0087	0.0903
80	0.0917	0.1003
90	0.0093	0.1076
100	0.0101	0.0923

In Table 2 below, we present the classification results after completely training the PNN on the entire dataset, reporting the FAR and FRR as a function of division values.

Table 2. FAR/FRR values as a function of the division level (the same values reported in Table 1). Note the values must be multiplied by 100 to give percentages. The values in the last row of the right-most columns are the averages of their respective columns.

Division points	False acceptance	False rejection
10	0.0483	0.0481
20	0.0192	0.0197
30	0.0576	0.0376
40	0.0576	0.0566
50	0.0576	0.0483
60	0.0001	0.0021
70	0.0576	0.0598
80	0.0481	0.0483
90	0.0288	0.0312
100	0.0480	0.0427
	0.0422	0.0394

The results presented in Table 2 indicate that our FAR/FRR is on the order of 4% - with a total error rate of 8.1% approximately.

5. Conclusions

We have successfully applied our modified Specht PNN to difficult biomedical datasets and have obtained accuracy levels comparable to other more traditional methods. In this study, we have employed our modified PNN to a small dataset of login id/password digraph samples. The modified classifier performed better than the standard PNN algorithm by approximately 20% (data not shown). Once trained, the classifier was able to minimise the FAR without significantly compromising the FRR (both were approximately 4%). These results are comparable to traditional neural network approaches as well as more 'modern' approaches such as SVM. The computational time required to train the PNN was minimal – on the order of 2 minutes on a standard Pentium IV desktop computer with low-to-typical amounts of memory (128 MB) and processing speed (1 GHz).

It must be noted that these results were obtained without any data pre-processing. We simply collected the data, selected a random subset for training (70%) and 30% for testing. This algorithm is time efficient when login id/password credentials are used for authentication purposes. It is a well known fact that the training phase of the PNN algorithm begins to degrade in terms of time efficiency when the sample numbers are large. But in this area of application, where we have a relatively small number of samples for training (on the order of 10-50) – and can select an equal number of testing samples, training performance is not an issue. This is in contrast to other techniques such as the backpropagation algorithm that requires a substantial number of training data in order to generate accurate classification. These advantages make the PNN a very suitable candidate for a novel machine learning algorithm in the context of keystroke dynamics authentication.

We will continue to pursue the use of PNNs for this particular domain - focusing on larger datasets to see how the training time scales with the number of users in the system. In addition, we may consider some unsupervised pre-processing of the data such as self-organised maps. It is important to remember that if this system is going to be used as an on-line authentication algorithm – data pre-processing must be kept to a minimum if it is to operate in an unsupervised manner – which may be a critical aspect of an on-line verification system. One must decide how much human effort must be spent in this process. Reduction of the pre-processing stage also tends to reduce the extent of overfitting the data.

5. References:

de Oliveira , M. VS, E. Kinto, Hernandez, E.D.M, & de Carvalho, T.C., User

Authentication Based on Human Typing Patterns with Artificial Neural Networks and Support Vector Machines, SBC 2005

de Ru, W.G. and Eloff, J.. "*Enhanced Password Authentication through Fuzzy Logic*". IEEE Expert, Vol.12, No.6, Nov/Dec, pp.38-45, 1997.

Hussien, B., Bleha, S. & McLaren, R., . *An application of fuzzy algorithms in a computer access security system*. Pattern Recognition Letters,9:39--43,1989.

Gaines, R. et al, Authentication by keystroke timing: Some preliminary results. Rand Report R-256-NSF. Rand Corp, 1980.

Gorunescu, F, Gorunescu, F,, El-Darzi, E, Gorunescu, S., & Revett K. A Cancer Diagnosis System Based on Rough Sets and Probabilistic Neural Networks, First European Conference on Health care Modelling and Computation, University of medicine and Pharmacy of Craiova, pp 149-159, 2005.

Jain, R. Bolle and Sharath Pankanti. "*Introduction to Biometrics*". In "Biometrics. Personal Identification in Networked Society". A.Jain, R.Bolle, S.Pankanti (Eds.). pp.1-41. Kluwer Academic Publishers., 2003

Joyce, R. and Gupta, G., Identity authorization based on keystroke latencies. *Communications of the ACM*. Vol. 33(2), pp 168-176, 1990.

Magalhães, S. T. and Santos, H. D., 2005, An improved statistical keystroke dynamics algorithm, *Proceedings of the IADIS MCCSIS 2005*.

Monrose, F. and Rubin, A. D., Authentication via Keystroke Dynamics. *Proceedings of the Fourth ACM Conference on Computer and Communication Security*. Zurich, Switzerland, 1997.

Monrose, F. et al, Password Hardening based on Keystroke Dynamics. *International Journal of Information Security*. 2001.

Obaidat, M.S. & Sadoun, B. A Simulation Evaluation Study of neural network techniques to Computer User Identification, Information Sciences 102, 239-258, 1997.

Peacock, A. et al, Typing Patterns: A Key to User Identification. *IEEE Security and Privacy*. September/October 2004.

Revett K. Magalhaes, S. & Santos, H., Developing a Keystroke Dynamics Based Agent Using Rough Sets, The 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshop on Rough Sets and Soft Computing in Intelligent Agents and Web Technology, Compiegne, France, 19-22 September, 2005

Revett,K., F. Gorunescu, M. Gorunescu, E. El-Darzi, M. Ene, *A Breast Cancer Diagnosis System: A Combined Approach Using Rough Sets and Probabilistic Neural Networks*, Proceedings Eurocon2005 –IEEE International Conference on "Computer as a tool", Belgrade, Serbia, November 21-24, pp, 1124-1127, 1-4244-0049-X/05/$20.00 ©2005 IEEE, 2005.

Specht,D.F. "Probabilistic neural networks for classification mapping or associative memory". Proceedings IEEE International Conference on Neural Networks, 1, 1988, 525-532.

Sung, K.S. & Cho S., GA SVM Wrapper Ensemble for Keystroke Dynamics Authentication, International Conference on Biometrics, Hong Kong, pp 654-660, 2006.

Tapiador, M. & Siguenza, J.A., Fuzzy Keystroke Biometrics on Web Security. AutoID '99 Proceedings. Workshop on Automatic Identification Advanced Technologies. IEEE. Pp 133-136., 1999.