COST-EFFECTIVE KERNEL RIDGE REGRESSION IMPLEMENTATION FOR KEYSTROKE-BASED ACTIVE AUTHENTICATION SYSTEM

Pei-Yuan Wu[1], *Chi-Chen Fang*[2], *J. Morris Chang*[2], *Stephen B. Gilbert*[2], *and S. Y. Kung*[1]

[1] Princeton University, [2] Iowa State University

ABSTRACT

In this study a keystroke-based authentication system is implemented on a large-scale free-text keystroke data set, where cost effective kernel-based learning algorithms are designed to enable trade-off between computational cost and accuracy performance. The authentication process evaluates the user's typing behavior on a vocabulary of words, where the judgments based on each word are concatenated by weighted votes, whose weights are also trained to provide optimal fusion of independent judgments. A novel truncated-RBF kernel is also implemented to provide better cost-performance trade-off. Experimental results validate the cost-effectiveness of the developed authentication system.

Index Terms— kernel methods, cost-effective, active authentication, keystroke, fusion methods, truncated-RBF

1. INTRODUCTION

The present username and password authentication system has many potential weaknesses [1, 2] such as password disclosure, easy-to-crack passwords, dictionary attacks, etc. The one-time log-in authentication system also suffers security weakness of an imposter gaining access to system resources by obtaining authenticated open sessions that is not properly monitored. Active authentication provides constant non-intrusive authentication by continuously monitoring user-specific physiological [3–5] and behavioral [6, 7] biometrics.

Scientists have noticed that neurophysiological factors involved in handwritten signatures also produce unique keystroke patterns [8], [9]. The study of monitoring keystroke dynamics as an additional layer of protection to the traditional password system has remained active since 1980s [2]. An excellent review and comparison of various keystroke-related algorithms is provided by Killourhy and Maxion's work [10].

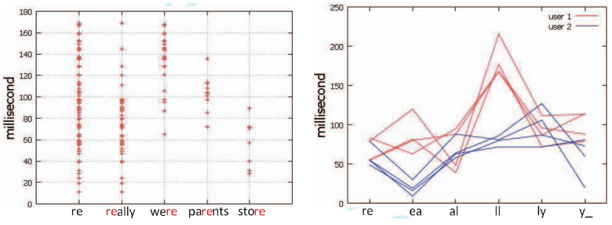
In this study a keystroke-based authentication system is implemented on a large-scale free-text keystroke data set collected by Chang et al. [11]. The large quantity of data raises issue on the cost-effectiveness in both learning and classification stages. This motivates our design of machine learning algorithms that enables cost-performance trade-off.

2. SYSTEM OVERVIEW

2.1. Cognitive Factors in Keystroke Dynamics

Conventional keystroke dynamics usually do not distinguish the timing difference between different words, but only consider a collection of digraphs. Fig.1(a) illustrates a collection of digraphs ("re") observed from the same user, but are collected from four different words: "really", "were", "parents", and "store". It shows that a user's typing behavior is not only dependent on digraphs, but also highly dependent on words.

To capture the cognitive factors, instead of breaking words into digraphs whose statistics are analyzed individually, we consider the whole word as one identity. Fig.1(b) illustrates the typing pattern of two users on the same word "really", which shows user-dependent characteristics.



(a) (b)

Fig. 1. (a) Digraph "re" from the same user in different words. (b) Two users typing the same word "really".

2.2. System Architecture

The authentication system is user-specific. The user profile of a legitimate user A contains a collection of most frequent vocabulary V_A, where each word $w \in V_A$ accompanies a classifier h_{Aw} that evaluates user A's keystroke typing pattern of word w. In continuous authentication process where an user B claims the identity of user A and types a word $w \in V_A$, the word classifier h_{Aw} gives a vote on whether or not user B should be authenticated as user A. The votes from all the words in V_A that user B types are then collected and weighted summed to arrive at the final decision. The details for determining the weights are discussed in Sec.5.

For a word w with M characters, the keystroke pattern is represented by a M-dimensional vector, where each element represents a time interval between consecutive characters (including the last space). The word classifier is then represented by a decision boundary in a M-dimensional space, which separates the positive region (accept as legitimate) and the negative region (reject as imposter).

3. LEARNING MODEL FORMULATION

To train the decision boundary of a word classifier h_{Aw} which summarizes user A's typing behavior on word w, we formulate a binary classification problem by partitioning all training samples of word w into two classes. The positive (legitimate) class comprises of samples collected from user A, while the negative (imposter) class is composed of samples from all other users.

Suppose word w has M characters and N samples available for training, the training data set can be represented as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^M$ is the feature vector and $y_i = \pm 1$ is the label, indicating the sample either belongs to the positive class ($y_i = +1$) or negative class ($y_i = -1$).

3.1. Kernel Methods

Kernel methods have long been popular tools in various supervised and unsupervised learning problems [12–16]. The basic insight is to nonlinearly transform patterns into some high-dimensional feature space, and then apply various linear pattern recognition methods.

By Mercer's Theorem [17], a kernel function that satisfies Mercer's condition can be represented as the inner product in a kernel-induced feature space $\mathcal{H}$, namely $k(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}$, where $\boldsymbol{\phi}(\mathbf{x})$ is some fixed mapping to $\mathcal{H}$. Common examples include Gaussian RBF kernel $k(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}$ and polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (1 + \frac{\mathbf{x}^T\mathbf{x}'}{\sigma^2})^p$.

3.2. Kernel Ridge Regression

Denote kernel-based regression function

$$h(\mathbf{x}) = \langle \mathbf{u}, \boldsymbol{\phi}(\mathbf{x}) \rangle_{\mathcal{H}} \tag{1}$$

The design objective for kernel ridge regression (KRR) [18–21] is to minimize the regulated empirical risk [22]:

$$\min_{\mathbf{u} \in \mathcal{H}} \sum_{i=1}^N L(h(\mathbf{x}_i), y_i) + \rho \|\mathbf{u}\|_{\mathcal{H}}^2 \tag{2}$$

By the linear space property [23, 24], the optimal solution to (2) is a linear combination of training inputs in $\mathcal{H}$:

$$\mathbf{u} = \sum_{i=1}^N a_i \boldsymbol{\phi}(\mathbf{x}_i) \tag{3}$$

By kernel trick, the regularization term can be rewritten as

$$\begin{aligned} \|\mathbf{u}\|_{\mathcal{H}}^2 &= \langle \sum_{i=1}^N a_i \boldsymbol{\phi}(\mathbf{x}_i), \sum_{j=1}^N a_j \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^N \sum_{j=1}^N a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{a}^T \mathbf{K} \mathbf{a} \end{aligned} \tag{4}$$

where $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix and $\mathbf{a} = [a_1 \ \cdots \ a_N]^T$. The regression function can be rewritten as a weighted sum of kernel functions

$$h(\mathbf{x}) = \langle \sum_{i=1}^N a_i \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle_{\mathcal{H}} = \sum_{i=1}^N a_i k(\mathbf{x}_i, \mathbf{x}) \tag{5}$$

Hence (2) can therefore be rewritten as

$$\min_{\mathbf{a} \in \mathbb{R}^N} \sum_{i=1}^N L(h(\mathbf{x}_i), y_i) + \rho \mathbf{a}^T \mathbf{K} \mathbf{a} \tag{6}$$

3.3. Class Dependent Costs for Imbalanced data set

Consider weighted squared error empirical risk

$$L(h(\mathbf{x}), y) = c(y)(h(\mathbf{x}) - y)^2 \tag{7}$$

where $c(y)$ is a positive class-dependent weight. The regularized empirical risk becomes

$$\begin{aligned} &\min_{\mathbf{a} \in \mathbb{R}^N} \sum_{i=1}^N c(y_i) \left(\sum_{j=1}^N a_j k(\mathbf{x}_j, \mathbf{x}_i) - y_i \right)^2 + \rho \mathbf{a}^T \mathbf{K} \mathbf{a} \\ &= \min_{\mathbf{a} \in \mathbb{R}^N} \|\mathbf{K}\mathbf{a} - \mathbf{y}\|_{\mathbf{C}}^2 + \rho \mathbf{a}^T \mathbf{K} \mathbf{a} \end{aligned} \tag{8}$$

where $\|\mathbf{r}\|_{\mathbf{C}}^2 = \mathbf{r}^T \mathbf{C} \mathbf{r}$ is the Mahalanobis norm, $\mathbf{C}$ is a diagonal matrix with $C_{ii} = c(y_i)$, and $\mathbf{y} = [y_1 \ \cdots \ y_N]^T$. Since (8) is convex and differentiable, it can be minimized by setting its derivative w.r.t. $\mathbf{a}$ equal to zero, giving the optimal solution

$$\mathbf{a} = (\mathbf{K} + \rho \mathbf{C}^{-1})^{-1} \mathbf{y} \tag{9}$$

Since the positive class contains only the legitimate user while the negative class contains all other users as imposters, the binary training data set is highly imbalanced in nature, where the positive class is outnumbered by the negative class. To avoid tendency for classifiers originally designed for balanced data sets to ignore the minorities and give poor results, we impose class-dependent costs and assign higher costs for misclassifying a positively-labeled sample. In this paper, the costs for misclassifying positive/negative samples are set to be inversely proportional to their population. More precisely, let N_+, N_- be the number of samples in positive/negative classes, respectively, we take

$$c(+1) = \frac{N}{2N_+}, \ \ c(-1) = \frac{N}{2N_-} \tag{10}$$

4. IMPROVING CLASSIFICATION COMPLEXITY OF KERNEL-BASED CLASSIFIERS

Based on our previous work [25] on cost-efficient KRR algorithms, our system enables trade-off between classification/learning complexity and accuracy performance by means of selecting appropriate finite decomposable kernel function.

4.1. Decision Function in Kernel Induced Feature Space

For finite decomposable kernel function, whose kernel-induced feature space $\mathcal{H} \subseteq \mathbb{R}^J$ has finite dimensions and Euclidean inner product

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{J} \phi^{(j)}(\mathbf{x})\phi^{(j)}(\mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}') \tag{11}$$

The regression function can be rewritten as

$$h(\mathbf{x}) = \sum_{i=1}^{N} a_i \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{u}^T \boldsymbol{\phi}(\mathbf{x}) \tag{12}$$

where the decision vector $\mathbf{u} = \sum_{i=1}^{N} a_i \boldsymbol{\phi}(\mathbf{x}_i)$ can be pre-computed in the learning phase.

Given a test pattern $\mathbf{x}$, it requires $O(J)$ operations to produce all elements of $\boldsymbol{\phi}(\mathbf{x})$, and another $O(J)$ operations to compute the inner product $\mathbf{u}^T \boldsymbol{\phi}(\mathbf{x})$. Therefore the total classification complexity is $O(J)$, which is independent of N.

In this paper, we consider p-th order polynomial kernel (abbreviated as POLY_p) $k(\mathbf{x}, \mathbf{x}') = (1 + \frac{\mathbf{x}^T\mathbf{x}'}{\sigma^2})^p$, whose basis functions take the following form

$$\phi^{(j)}(\mathbf{x}) = \sqrt{\frac{p!}{(p-\ell)!}} \prod_{m=1}^{M} \frac{1}{\sqrt{d_m!}} \left(\frac{x^{(m)}}{\sigma}\right)^{d_m} \tag{13}$$
$$0 \le \ell \le p, \ell = d_1 + \cdots + d_M$$

There are $J = J^{(p)} = \frac{(M+p)!}{M!p!}$ different combinations.

The flexibility in classification schemes (cf. (5,(12)) results in classification complexity $O(\min(NM, J))$, which is especially useful in this study: Since users tend to type short words more often than long words, we may apply (12) to short words which have many samples but relatively small kernel-induced feature space dimension J; On the contrary, (5) is suitable for long words, as they have less learning samples but relatively large J.

4.2. Finite J-Degree Approximation of RBF Kernel

We introduce a truncated-RBF (TRBF) kernel [25]

$$\begin{aligned} k_{TRBF}(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \left(\sum_{\ell=1}^{p} \frac{1}{\ell!} \left(\frac{\mathbf{x}^T\mathbf{x}'}{\sigma^2}\right)^{\ell}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{2\sigma^2}\right) \\ &= \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}') \end{aligned} \tag{14}$$

where each basis function takes the following form

$$\phi^{(j)}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \prod_{m=1}^{M} \frac{1}{\sqrt{d_m!}} \left(\frac{x^{(m)}}{\sigma}\right)^{d_m} \tag{15}$$
$$0 \le d_1 + \cdots + d_M \le p$$

The trade-off between accuracy performance and computation efficiency highly depends on order p and its intrinsic dimension $J^{(p)}$, which is identical to that of polynomial kernels. In this paper we refer to polynomial/TRBF kernels with order p as POLY_p and TRBF_p, respectively. Note that TRBF is simply a Taylor expansion approximation of RBF. For a more sophisticated RBF approximation, see [26].

4.3. Fast Learning Kernel Methods

For finite decomposable kernel function (cf. (11)), the kernel matrix is tightly linked to the training inputs in $\mathcal{H}$:

$$\mathbf{K} = \boldsymbol{\Phi}^T \boldsymbol{\Phi} \tag{16}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1) \cdots \boldsymbol{\phi}(\mathbf{x}_N)]$ is the data matrix in kernel-induced feature space.

4.3.1. Learning Complexity of SVM

The SVM learning involves a quadratic programming problem, whose learning complexity is reportedly to grow at a modest rate near $O(N^2)$.

4.3.2. Learning Complexity for KRR

The KRR learning focuses on solving the decision vector $\mathbf{a}$ in (9), which involves inverting a $N \times N$ matrix $(\mathbf{K} + \rho\mathbf{C}^{-1})$ and therefore demands a high complexity of $O(N^3)$.

For incremental learning scenario where the training samples are collected indefinitely with time, N will eventually become too large for both KRR and SVM to be computationally affordable. Several methods were proposed to relieve computation burden [19, 27, 28]. In this work we propose a cost-efficient algorithm whose learning complexity grows linearly with N, as described as below.

4.3.3. Fast Algorithm for KRR

By Woodbury's inversion lemma, one has

$$\begin{aligned} (\mathbf{K} + \rho\mathbf{C}^{-1})^{-1} &= (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \rho\mathbf{C}^{-1})^{-1} \\ &= \rho^{-1}\mathbf{C} - \rho^{-1}\mathbf{C}\boldsymbol{\Phi}^T(\rho\mathbf{I} + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^T)^{-1}\boldsymbol{\Phi}\mathbf{C} \end{aligned} \tag{17}$$

In other words, the decision vector has an explicit form

$$\begin{aligned} \mathbf{u} = \boldsymbol{\Phi}\mathbf{a} &= \boldsymbol{\Phi}(\mathbf{K} + \rho\mathbf{C}^{-1})^{-1}\mathbf{y} \\ &= (\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^T + \rho\mathbf{I})^{-1}\boldsymbol{\Phi}\mathbf{C}\mathbf{y} \end{aligned} \tag{18}$$

The fast-KRR algorithm solves decision vector $\mathbf{u}$ instead of $\mathbf{a}$, which incurs three costs: (1) The computation of the $J \times J$ matrix $\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^T$ requires $O(J^2N)$ operations; (2) The inversion of $(\boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^T + \rho\mathbf{I})$ requires $O(J^3)$ operations; (3) The matrix-vector multiplication requires a negligible $O(NJ)$ operations. In summary, the learning complexity is $O(J^3 + J^2N)$, which is linear w.r.t. N.

5. FUSION METHODS

In Chair et al.'s work [29], a fusion scheme is proposed which combines decisions from multiple independent classifiers by weighted votes. The weights depend not only on the classifier, but also on its outcome. The baseline is that information provided by acceptance or rejection is not equal and is dependent on the classifier's FRR and FAR. Intuitively speaking, for a classifier with very low FRR but rather moderate FAR, since false rejection is more unlikely than false acceptance, its rejection votes would have larger weights compared to acceptance votes. On the other hand, for a classifier with moderate

Table 1. Accuracy and computational cost comparison.

Kernel	Train / Test time	Kernel	Train / Test time
linear	16.3 / 14.8 sec		
POLY2	35.9 / 18.2 sec	TRBF2	35.6 / 18.0 sec
POLY3	101.2 / 21.7 sec	TRBF3	98.2 / 20.3 sec
POLY4	491.6 / 28.0 sec	TRBF4	486.0 / 28.0 sec

FRR but very low FAR, its acceptance votes should be more persuasive than rejection votes.

Following their concepts, in this study there are two weights accompanying with each word classifier h_{Aw}, namely the acceptance weight $\beta_{Aw}^{(acc)}$ and the rejection weight $\beta_{Aw}^{(rej)}$. Both weights are determined by the estimated FAR and FRR performances $\hat{p}_{FAR}$, $\hat{p}_{FRR}$ as follows

$$\beta_{Aw}^{(acc)} = \log\left(\frac{1-\hat{p}_{FRR}}{\hat{p}_{FAR}}\right), \quad \beta_{Aw}^{(rej)} = \log\left(\frac{1-\hat{p}_{FAR}}{\hat{p}_{FRR}}\right) \tag{19}$$

The authentication process maintains a confidence score $s_{BA}(T)$ representing how confident the system is to authenticate user B as user A at time stamp T. If user B types word w at time stamp T, the confidence score is updated as

$$s_{BA}(T) = \begin{cases} s_{BA}(T-1) + \beta_{Aw}^{(acc)} & \text{(accept)} \\ s_{BA}(T-1) - \beta_{Aw}^{(rej)} & \text{(reject)} \end{cases} \tag{20}$$

In this study, the FAR and FRR performances of all word classifiers are evaluated by 3-fold cross validation.

6. EXPERIMENT

To verify the cost-performance trade-off, we conduct experiments on free-text keystroke data set collected by Chang et al. [11]. Table.1 and Fig.2 summarize the computational costs as well as prediction accuracies for various polynomial and TRBF kernels. Our result is consistent with Chang et al.'s work [11]. Note that although KRR with TRBF2 kernel is slightly inferior in terms of accuracy performance than SVM with Gaussian-RBF kernel, it enjoys a major advantage in terms of much faster learning and classification speeds. The result also shows a consistent trade-off between learning/classification computational cost and accuracy performance: With higher order kernels and thus stronger representation power, smaller EER is achieved with higher computational cost. In numbers, TRBF_4 kernel yields 3.2% EER while TRBF_2 kernel yields slightly inferior 4.05% EER but provides a 10-fold saving in training cost.

The accuracy performance of polynomial kernels saturates at POLY_3, possibly due to overfitting by high-order terms. On the other hand, TRBF allows further error reduction. This observation suggests that TRBF is a more appropriate choice of finite decomposable kernel for cost-performance trade-off. In fact, TRBF_p always yields better accuracy performance than POLY_p kernels, even though they have similar forms and the same induced feature space dimension $J^{(p)}$.

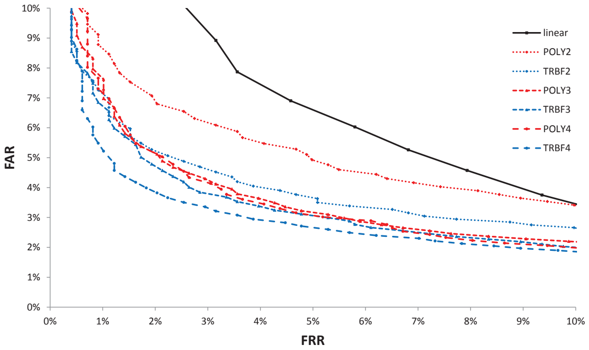


Fig. 2. Detection error rates of polynomial and TRBF kernels.

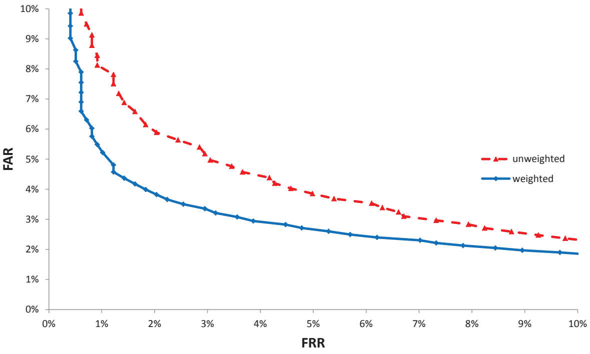


Fig. 3. Comparison between weighted and unweighted voting schemes for TRBF4 kernel.

Since TRBF_p imposes less weights on high order terms than POLY_p kernels, it has smaller high-order term components in the induced feature space $\mathcal{H}$, which are further suppressed by the regularization mechanism [24, Ch.9.8]. This alleviates the overfitting problem commonly suffered by high-order polynomial kernels. In Fig.3 the accuracy performance of weighted votes (cf. (20)) versus unweighted votes (fix $\beta_{Aw}^{(acc)} = \beta_{Aw}^{(rej)} = 1$) are compared, which shows apparent improvement with adequately learned weights.

In summary, the proposed TRBF kernel enables tradeoff between learning/classification costs and prediction accuracy. Hopefully, this work provides a proof of concept of a feasible kernel approach to keystroke-based active authentication systems. In the future, we shall explore e.g., (1) combining other low rank approximation methods (e.g., [19, 27, 28]) and (2) fine-tuning the parameters (e.g., σ) to further improve the order-performance trade-off.

Acknowledgment

This work was supported by the Active Authentication Program of DARPA under grant FA8750-12-2-0200.[1]

[1] The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

7. REFERENCES

[1] Anne Adams and Martina Angela Sasse, "Users are not the enemy," *Commun. ACM*, vol. 42, no. 12, pp. 41–46, 1999.

[2] Alen Peacock, Xian Ke, and Matthew Wilkerson, "Typing patterns: A key to user identication," *IEEE Secur. Privacy Mag.*, vol. 2, no. 5, pp. 40–47, 2004.

[3] Koichiro Niinuma, Unsang Park, and Anil K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 771–780, 12 2010.

[4] John Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 2004, 1 2004.

[5] Terence Sim, Sheng Zhang, Rajkumar Janakiraman, and Sandeep Kumar, "Continuous verification using multimodal biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.,*, vol. 29, no. 4, pp. 687–700, 4 2007.

[6] Daniele Gunetti and Claudia Picardi, "Keystroke analysis of free text," *ACM Trans. Inf. Syst. Secur*, vol. 8, no. 3, pp. 312–347, 8 2005.

[7] Maja Pusara and Carla E. Brodley, "User re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, New York, NY, USA, 2004, VizSEC/DMSEC '04, pp. 1–8, ACM.

[8] Rick Joyce and Gopal Gupta, "Identity authentication based on keystroke latencies," *Commun. ACM*, vol. 33, no. 2, pp. 168–176, 2 1990.

[9] R. Spillane, "Keyboard apparatus for personal identification," *IBM Technical Disclosure Bulletin*, vol. 17, no. 3346, 1975.

[10] Kevin S. Killourhy and Roy A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proceedings of the 39th Annual International Conference on Dependable Systems and Networkds*, Los Alamitos, 2009, DSN 2009, pp. 125–134, IEEE Computer Society Press.

[11] J. Morris Chang, Chi Chen Fang, Kuan Hsing Ho, Norene Kelly, Peiyuan Wu, Yixiao Ding, Chris Chu, Stephen Gilbert, Amed E. Kamal, and Sun Yuan Kung, "Capturing cognitive fingerprints from keystroke dynamics," *IT Professional*, vol. 15, no. 4, pp. 24–28, July-August 2013.

[12] Federico Girosi, Michael Jones, and Tomaso Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, pp. 219–269, 1995.

[13] V. N. Vapnik, *The Nature of Statistical Learning Theorey*, Springer-Verlag, New York, 1995.

[14] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[15] Michael E. Mavroforakis and Sergios Theodoridis, "A geometric approach to support vector machine (svm) classification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 671–682, 5 2006.

[16] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, Wiley, New York, 2 edition, 2001.

[17] J. Mercer, "Functions of postivie and negative type, and their connection with the theory of integral equations," *Trans. of the London Philosophical Society (A)*, vol. 209, pp. 415–446, 1909.

[18] Thilo-Thomas Frieb and Robert F. Harrison, "A kernel-based adaline," in *ESANN 1999, 7th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 21-23, 1999, Proceedings*, 1999, pp. 245–250.

[19] Yaakov Engel, Shie Mannor, and Ron Meir, "The kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, 2003.

[20] Jyrki Kivinen, Alex J. Smola, and Robert C. Williamson, "Online learning with kernels," *IEEE Trans. on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 8 2004.

[21] Weifeng Liu, Puskal P. Pokharel, and Jose C. Principe, "The kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, 2 2008.

[22] A. N. Tychonoff, "On the stability of inverse problems," *Doklady Akademii Nauk SSSR*, vol. 39, no. 5, pp. 195–198, 1943.

[23] M. A. Aizerman, E. A. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," in *Automation and Remote Control,*, 1964, number 25 in Automation and Remote Control,, pp. 821–837.

[24] S. Y. Kung, *Kernel Methods and Machine Learning*, Cambridge University Press, 2014.

[25] S. Y. Kung and P. Wu, "On efficient learning and classification kernel methods," *Proceeding of ICASSP, Kyoto*, March 2012.

[26] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 480–492, March 2012.

[27] S. Phonphitakchai and T.J. Dodd, "Stochastic meta descent in online kernel methods," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, May 2009, vol. 02, pp. 690–693.

[28] C. Richard, J.C.M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *Signal Processing, IEEE Transactions on*, vol. 57, no. 3, pp. 1058–1067, March 2009.

[29] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerospace Election. Sys.*, vol. 22, pp. 98–101, Jan 1986.