Pattern Recognition Letters 32 (2011) 1070–1080

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

On the discriminability of keystroke feature vectors used in fixed text keystroke authentication

Kiran S. Balagani [a,*], Vir V. Phoha [a], Asok Ray [b], Shashi Phoha [c]

[a] Computer Science, Louisiana Tech University, LA 71270, USA
[b] Mechanical and Nuclear Engineering, Penn State University, PA 16802, USA
[c] Electrical and Computer Engineering, Penn State University, PA 16802, USA

ARTICLE INFO

Article history:
Received 29 June 2009
Available online 16 February 2011
Communicated by S. Sarkar

Keywords:
Keystroke dynamics
User recognition
Feature selection analysis
Bhattacharya distance
Parzen window
Mahalanobis distance

ABSTRACT

Heterogeneous and aggregate vectors are the two widely used feature vectors in fixed text keystroke authentication. In this paper, we address the question "*Which vectors, heterogeneous, aggregate, or a combination of both, are more discriminative and why?*" We accomplish this in three ways – (1) by providing an intuitive example to illustrate how aggregation of features inherently reduces discriminability; (2) by formulating "discriminability" as a non-parametric estimate of Bhattacharya distance, we show theoretically that the discriminability of a heterogeneous vector is higher than an aggregate vector; and (3) by conducting user recognition experiments using a dataset containing keystrokes from 33 users typing a 32-character reference text, we empirically validate our theoretical analysis. To compare the discriminability of heterogeneous and aggregate vectors with different combinations of keystroke features, we conduct feature selection analysis using three methods: (1) ReliefF, (2) correlation based feature selection, and (3) consistency based feature selection. Results of feature selection analysis reinforce the findings of our theoretical analysis.

1. Introduction

Two types of keystroke feature vectors are widely used in fixed text authentication systems: (1) heterogeneous vectors, which have both key hold and key interval latencies as constituent features and (2) aggregate vectors, which can have key press latencies, key release latencies, or trigraphs as constituent features. Obaidat and Sadoun (1997) *empirically* showed that feature vectors with key hold and key interval latencies combined and Robinson et al. (1998) *empirically* showed that feature vectors with key hold and key press latencies combined resulted in better authentication performance than with vectors made from key press or key hold latencies alone. Both studies explained the performance improvement by stating that combining features increases the dimensionality of a feature vector, which may result in better discrimination of users.

In spite of the preliminary evidence on the advantage of combining keystroke features, the composition of feature vectors used for fixed text authentication has been quite ad hoc in the literature. As an example, in Table 1, we list nine fixed text authentication studies that have appeared during 2000–2009. A scan through the column 'Feature Vector Type' reveals how the feature vector composition in each study differs, ranging from vectors containing key press latencies alone to vectors containing combinations of key hold, key interval, key press, and key release latencies. Surprisingly, only three out of the nine studies provided empirical justifications for their choice of the feature vector. The studies are: (1) Bergandano et al. (2002), justified their choice of using a vector of trigraphs by showing that trigraphs had the best mean intra-user and inter-user distances compared to digraphs or 4-graphs for their data; (2) Yu and Cho (2003), demonstrated that key press and key interval latencies selected using a wrapper feature selection technique reduced the average false accept rate (FAR) from 15.78% (with all key hold and key interval latencies) to 3.54% (with only three selected latencies) at 0% false reject rate (FRR); and (3) Hosseinzadeh and Krishnan (2008), tested their Guassian Mixture Model method with feature vectors constituting key press latencies, key release latencies, key hold latencies, and their combinations. They showed that the best performance (4.3% FAR at 4.8% FRR) was achieved using a vector of key hold and key release latencies.

Considering the existing disparity in the types of feature vectors used in fixed text authentication literature, one question immediately arises – *What type of feature vector, heterogeneous, aggregate, or a combination of both, is good and why?* In this paper, we address the question from three standpoints: (1) intuitive standpoint – using an example, we illustrate how the aggregation of features

* Corresponding author. Address: Center for Secure Cyberspace, Computer Science, Louisiana Tech University, Ruston, LA 71270, USA. Tel.: +1 318 514 9745; fax: +1 318 257 4922.
E-mail address: ksb011@latech.edu (K.S. Balagani).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.02.014

Table 1
Nine studies in fixed text user authentication. The second column indicates the type of feature vector used in each study.

Paper	Feature Vector Type
Haider et al. (2000)	Aggregate (key press)
Bergandano et al. (2002)	Aggregate (trigraphs)
Yu and Cho (2003)	Heterogeneous (key hold and key interval)
Sheng et al. (2005)	Heterogeneous (key hold and key interval)
Revett et al. (2005)	Aggregate (key press)
Arajo et al. (2005)	Heterogeneous and aggregate (key press, key hold, key interval)
Filho and Freire (2006)	Aggregate (key press)
Hosseinzadeh and Krishnan (2008)	Heterogeneous and aggregate (key hold, key release, and key press)
Giot et al. (2009)	Heterogeneous and aggregate (key press, key hold, key interval, key release)

effects "discriminability" (i.e., the ability of a feature vector to distinguish one user from another); (2) theoretical standpoint – we show theoretically why heterogeneous vectors are preferable over aggregate vectors; and (3) empirical standpoint – we validate our theoretical analysis through classification experiments and use feature selection analysis to compare the performance of combinations of features with heterogeneous and aggregate vectors. The contributions of our paper follow.

1.1. Contributions of our paper

We present a *theoretical analysis* to show how heterogeneous vectors can have higher discriminability than aggregate vectors. For our theoretical analysis, we formulate the discriminability of a keystroke vector as a kernel density estimate of Bhattacharya distance between class-conditional densities. This formulation avoids making explicit assumptions on the distributions of class-conditional densities, allows a realistic representation of discriminability as the "geometric" distance between the users' samples, and facilitates an analysis that is independent of the underlying recognition method.

We validate our theoretical analysis with user recognition experiments on a keystroke dataset obtained from 33 users typing a fixed 32-character reference text. We used four classifiers: (1) naive Bayes, (2) tree augmented naive Bayes, (3) k-nearest neighbor, and (4) ridge logistic regression for user recognition. Results clearly demonstrate the superiority of heterogeneous vectors over aggregate vectors.

We perform *feature selection analysis* to compare the performance of different combinations of features with heterogeneous and aggregate vectors. We use three state-of-the-art feature selection methods: (1) ReliefF, (2) correlation based selection, and (3) consistency based selection. Results show that heterogeneous vector performs better than different combinations of features selected by the feature selection methods. In contrast, several combinations of features outperform an aggregate vector.

1.2. Overall gains our work brings to fixed text authentication field

Our work addresses a fundamental question – *Which feature vector, heterogeneous, aggregate, or a combination of both, is more discriminative?*

One way to address this question is to identify the best performing feature vectors by empirically evaluating feature vectors with different keystroke authentication methods. The problem with this approach is that the performance of feature vectors can vary widely across methods. For example, using the same feature vector under the same evaluation conditions, Killourhy and Maxion (2009) achieved the best equal error rate (0.096%) with a Manhattan distance based method and the worst equal error rate (0.829%) with a neural network (i.e., though the feature vector remained same, the performance worsened 9.19 times when the method changed). Therefore, conclusions based entirely on empirical results are more reflective of how well "the methods" exploit the feature vector rather than being reflective of the intrinsic ability of the feature vector to seperate users.

In contrast, in our theoretical analysis, we formulate the "discriminability" of a feature vector as the Mahalanobis distance between the samples of different users. Because Mahalanobis distance is a measure of the geometric distance between data samples, our formulation of discriminability closely reflects how well a feature vector "physically" separates samples of different users. Also, the insights gained from our theoretical analysis do not fluctuate between methods.

Our work advances (Obaidat and Sadoun, 1997; Robinson et al., 1998) in two ways:

1. Obaidat and Sadoun (1997) and Robinson et al. (1998) explained the performance improvement of the combination of features by stating that combining features increases the dimensionality of a feature vector, which may result in better discrimination of users. While the argument that higher dimensionality may provide higher discriminability is not necessarily true, the reason why heterogeneous feature vectors perform better than aggregate vectors is that aggregation (e.g., adding key hold and key interval latencies to form key press latencies) may render two samples previously separable in the heterogeneous feature space to become inseparable in the aggregated feature space. We clarify this reasoning with an intuitive example and with the results of our feature selection analysis.
2. The argument that higher dimensionality may provide higher discriminability is in itself incomplete without considering the correlations (i.e., dependencies) between features. In our theoretical analysis, we compare the discriminability between heterogeneous and aggregate vectors assuming two types of correlation structures. The first structure assumes equal correlation among features. The second correlation structure assigns higher correlation values to keystroke features of adjacent characters. Under the second correlation structure, the difference in the discriminability between heterogeneous and aggregate vectors increases with the number of characters in the reference text. Under the first correlation structure, the difference in the discriminability between heterogeneous and aggregate vectors peaks between 5 and 30 characters and then decreases gradually, suggesting that heterogeneous vectors perform better with short (password and sentence type) reference texts. Our empirical results further validate this point.

1.3. Organization of the rest of the paper

In Section 2 we review keystroke feature vectors. In Section 3 we present an intuitive example and theoretical analysis to compare the discriminability of heterogeneous and aggregate vectors. In Section 4 we present user recognition experiments and feature selection analysis; and we discuss results. We conclude our work in Section 5.

2. Review of keystroke feature vectors

Fig. 1 illustrates the key press (P_t, P_h, P_e) and key release (R_t, R_h, R_e) times obtained when the phrase '**the**' is typed. The following keystroke features can be calculated from these times: (1) *key hold* latencies of the letters 't', 'h', and 'e' are calculated as

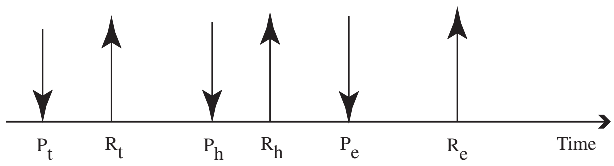

Fig. 1. Key press (P_t, P_h, P_e) and key release times (R_t, R_h, R_e) obtained when the word '**the**' is typed on the keyboard.

$KH_t = R_t - P_t$, $KH_h = R_h - P_h$, and $KH_e = R_e - P_e$ respectively, (2) *key interval* latencies of the letter pairs 'th' and 'he' are calculated as $KI_{th} = P_h - R_t$ and $KI_{he} = P_e - R_h$, respectively, (3) *key press* latencies of 'th' and 'he' are calculated as $KP_{th} = P_h - P_t$ and $KP_{he} = P_e - P_h$, and (4) *trigraph* latency of the letters 'the' is calculated as $T_{the} = P_e - P_t$.

A keystroke feature vector is **homogeneous** if it has either key hold latencies *only* or key interval latencies *only* as its features. A homogeneous vector is the simplest of all keystroke vectors and is undecomposable. The vectors (KH_t, KH_h, KH_e) and (KI_{th}, KI_{he}) are homogeneous vectors. A feature vector is **heterogeneous** if it has *both* key hold and key interval latencies as features. The vector $(KH_t, KI_{th}, KH_h, KI_{he}, KH_e)$ is a heterogeneous vector. A feature vector is an **aggregate** vector if its features are derived by aggregating key hold and key interval latencies. A vector of key press latencies (KP_{th}, KP_{he}) is an aggregate vector because it is formed by *adding* key hold and key interval latencies as $KP_{th} = KH_t + KI_{th}$ and $KP_{he} = KH_h + KI_{he}$. A vector of trigraphs, 4-graphs, or n-graphs is also an aggregate vector.

Let 'd' be the number of characters in the reference text. The size of a homogeneous vector generated by typing the reference text is at most d with key press latencies and is at most $(d - 1)$ with key interval latencies. The size of a heterogeneous vector is at most $(2d - 1)$. The size of an aggregate vector is at most $(d - 1)$ with key press latencies and $(d - 2)$ with trigraphs.

3. Discriminability in heterogeneous and aggregate vectors

In this section, we demonstrate how the aggregation of features may decrease discriminability (i.e., class separability). Let x and y represent two real-valued features; (x, y) represent the joint feature space; and $z = x + y$.

Fig. 2(a) shows a hypothetical XOR type classification problem (Duda et al., 2000) in which samples were generated using Normal distributions with mean vectors [1.5, 1.5] and [7.5, 7.5] for Class I and [1.5, 7.5] and [7.5, 1.5] for Class II and identity covariance

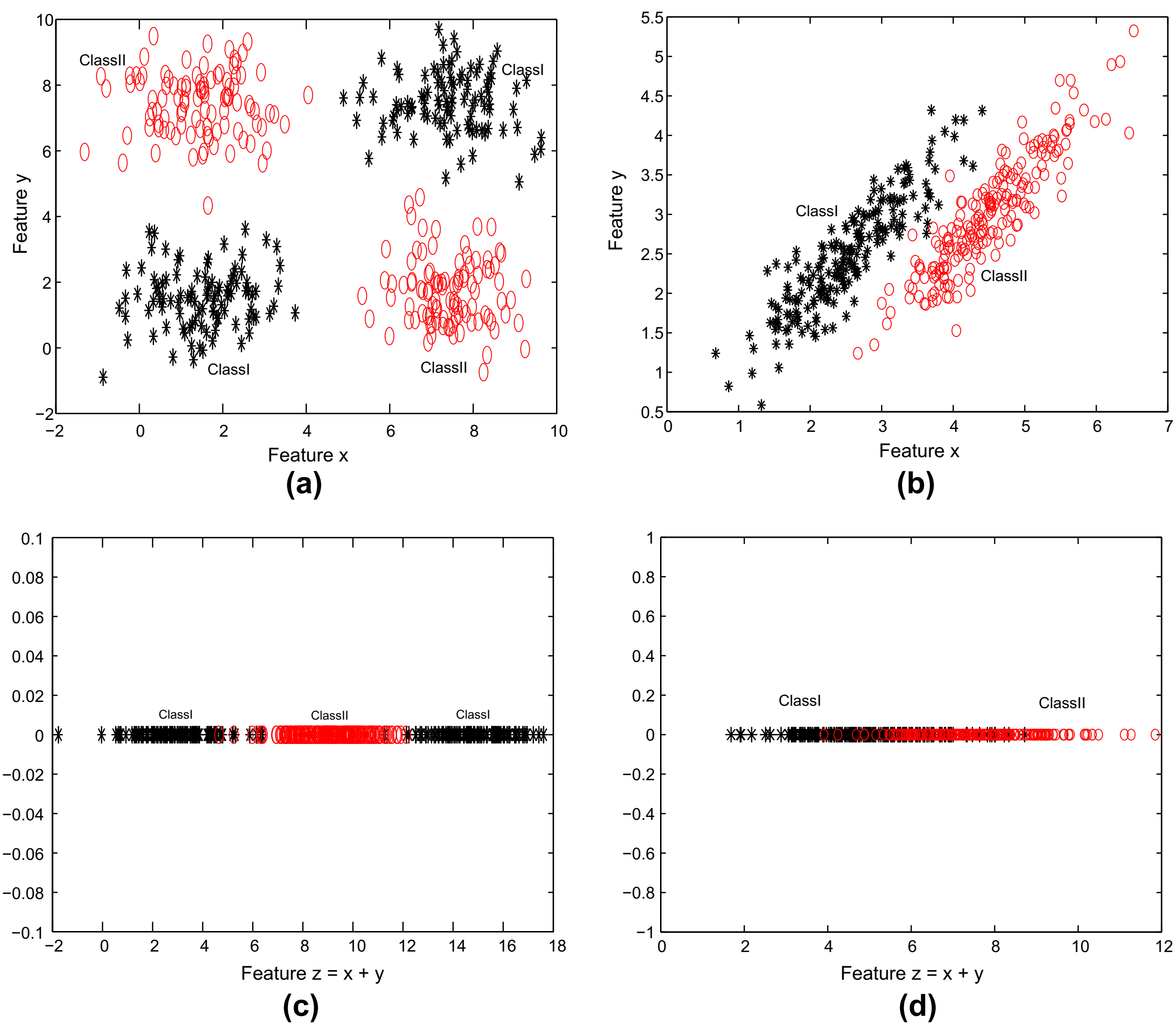

Fig. 2. (a) Two classes occurring as well separated clusters in (x, y) joint feature space; (b) two highly correlated but well-separated classes in (x, y); (c) shows how the samples of the two classes in (a) overlap when feature space is changed to a one-dimensional aggregate space $z = x + y$; and (d) shows how the samples of (b) overlap in z feature space.

matrices. In Fig. 2(a), though the densities of the two classes heavily overlap when the features x and y are considered individually, the classes are well separated in the (x,y) joint feature space. Fig. 2(b) plots samples generated using Normal distribution with mean vectors [2.5,2.5] for Class I, [4.5,3.0] for Class II, and a common covariance matrix [0.5 0.45; 0.45 0.5]. The correlation coefficient between x and y is 0.9. Fig. 2(b) illustrates the point that classes can be well-separated in (x,y) joint feature space even when x and y are highly correlated.

Fig. 2(c) and (d) plots the samples in Fig. 2(a) and (b) respectively on a one-dimensional feature space $z = x + y$. Fig. 2(c) and (d) illustrate how the classes that were well separated in (x,y) joint feature space overlap in z feature space, indicating that the (x,y) joint feature space may have higher discriminability than the z feature space.

Because heterogeneous feature space is a *joint* feature space of key hold and key interval latencies, whereas an aggregate feature space is a feature space of key press latencies (or trigraphs) formed by *adding* key hold and key interval latencies, the addition of features may reduce discriminability, as illustrated through Fig. 2 Next, we present a theoretical analysis to compare the discriminability of heterogeneous and aggregate vectors.

3.1. Theoretical analysis

We present a theoretical analysis on the discriminability of heterogeneous and aggregate keystroke vectors using an estimate of Bhattacharya distance. Let ω_1 and ω_2 be two classes with prior probabilities $p(\omega_1)$ and $p(\omega_2)$, respectively. Let X be a d-dimensional feature vector in $\Re^d$. Let $p(X|\omega_1)$ and $p(X|\omega_2)$ be the conditional densities. Kailath (1967) showed that the Bayes classification error is upper bounded by $p(\omega_1)p(\omega_2)\ B_X$, where $B_X = \int \sqrt{p(X|\omega_1)p(X|\omega_2)}dX$ is the Bhattacharya distance between the classes in the feature space X. Assuming equal priors, a higher B_X corresponds to higher discriminability (or separability) between classes ω_1 and ω_2 and lower classification error. In the following section, we generalize the result of Heydorn (1967) to show that B_X is the Mahalanobis distance between the samples from the classes ω_1 and ω_2. For simplicity, we consider only two classes. Extension to the multi-class case is straight forward using the formulation in (Lainiotis, 1969).

3.1.1. Bhattacharya distance with Parzen estimates

Let $\{X_1,\ldots,X_n\} \in \omega_1$ and $\{X'_1,\ldots,X'_n\} \in \omega_2$ be d-dimensional training vectors. Let the Parzen window estimates of conditional densities be

$$p(X|\omega_1) \approx \hat{p}(X|\omega_1) = \frac{1}{n}\sum_{i=1}^{n}\phi(X - X_i, h_1) \quad \text{and} \quad p(X|\omega_2) \approx \hat{p}(X|\omega_2) = \frac{1}{n}\sum_{i=1}^{n}\phi(X - X'_i, h_2),$$

where the window function obeys $\phi(X,h) \geqslant 0$ and $\int \phi(X,h)dX = 1$. The width parameter h is a function of n such that $\lim_{n\to\infty} h(n) = 0$ and $\lim_{n\to\infty} nh^d(n) = 0$. Parzen (1962) showed that $\hat{p}(X|\omega)$ converges to the true density $p(X|\omega)$ if $\phi(X,h)$ and h are chosen properly. By assuming the window function to be a d-dimensional Gaussian,[1] the result of Heydorn (1967) can be generalized with arbitrary covariance matrices to show that the Bhattacharya distance B_X is proportional to

$$R = \left(\frac{1}{8nh_1h_2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}(X_i - X'_j)^T\left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1}(X_i - X'_j) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}, \tag{1}$$

where "R" represents the discriminability or the degree of separation between the samples $\{X_1,\ldots,X_n\}$ of class ω_1 and the samples $\{X'_1,\ldots,X'_n\}$ of class ω_2, Σ_1 represents the covariance matrix of class ω_1, and Σ_2 represents the covariance matrix of class ω_2. We choose to estimate Bhattacharya distance with Parzen window estimates to avoid making restrictive assumptions that the conditional probabilities $P(X|\omega_1)$ and $P(X|\omega_2)$ originate from specific distributions. By assuming $\Sigma_1 = \Sigma_2 = \Sigma$, R in (1) becomes the Mahalonobis distance between the training vectors of classes ω_1 and ω_2. Because it is difficult to compute R for an arbitrary covariance matrix, we follow the approach taken by Jain and Waller (1978) and assume that Σ is a Toeplit matrix (Gray, 2006). In our analysis, we consider two Toeplitz matrices:

$$A = \begin{bmatrix} 1 & \rho & \rho & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \rho & \rho & 1 & \cdots & \rho \\ \vdots & & & \ddots & \vdots \\ \rho & \rho & \rho & \cdots & 1 \end{bmatrix}_{d\times d} \quad \text{and}$$

$$B = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{d-1} \\ \rho & 1 & \rho & \cdots & \rho^{d-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{d-3} \\ \vdots & & & \ddots & \vdots \\ \rho^{d-1} & \rho^{d-2} & \rho^{d-3} & \cdots & 1 \end{bmatrix}_{d\times d}, \tag{2}$$

where ρ is the correlation between features. (See Jain and Waller (1978) for the inverse of A and B.)

If $\Sigma = A$ (i.e., assuming all keystroke features have the same correlation), (1) becomes

$$R_{A,d} = \left(\frac{1}{8nh_1h_2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_d\sum_{k=1}^{d}\left(x_{ik} - x'_{jk}\right)^2 + 2\beta_d\sum_{\substack{m=1:d\\ l=m+1:d}}\left(x_{im} - x'_{jm}\right)(x_{il} - x'_{jl})\right), \tag{3}$$

where $R_{A,d}$ is the discriminability between classes ω_1 and ω_2 when $\Sigma = A$ and d is the dimensionality of the vectors X_i and X'_j, $(x_{i1}, x_{i2},\ldots,x_{id})$ and $\left(x'_{j1}, x'_{j2},\ldots,x'_{jd}\right)$ are the features of vectors X_i and X'_j respectively, $\alpha_d = \frac{1+(d-2)\rho}{1+(d-2)\rho-(d-1)\rho^2}$ and $\beta_d = \frac{-\rho}{1+(d-2)\rho-(d-1)\rho^2}$. If $\Sigma = B$ (i.e., assuming that the correlation between keystroke features from adjacent characters in the reference phrase is higher than the correlation between keystroke features from non-adjacent characters), (1) becomes

$$R_{B,d} = \left(\frac{1}{8nh_1h_2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha\sum_{k=1,d}\left(x_{ik} - x'_{jk}\right)^2 + \gamma\sum_{k=2}^{d-1}\left(x_{ik} - x'_{jk}\right)^2 + 2\beta\sum_{\substack{m=1:d\\ l=m+1}}\left(x_{im} - x'_{jm}\right)\left(x_{il} - x'_{jl}\right)\right), \tag{4}$$

where $R_{B,d}$ is the discriminability between classes ω_1 and ω_2 when $\Sigma = B$ and d is the dimensionality of the vectors X_i and X'_j, $(x_{i1}, x_{i2},\ldots,x_{id})$ and $\left(x'_{j1}, x'_{j2},\ldots,x'_{jd}\right)$ are the features of vectors

[1] With enough samples, Parzen window estimates converge to arbitrary target densities, irrespective of the choice of the window function $\phi(\cdot)$ and the width parameter h (see Duda et al., 2000).

X_i and X'_j respectively, $\alpha = \frac{1}{1-\rho^2}, \gamma = \frac{1+\rho^2}{1-\rho^2}$, and $\beta = \frac{-\rho}{1-\rho^2}$. Jain and Waller (1978) calculated α_d, β_d, α, γ, and β.

In the following section, we use (3) and (4) to compare the discriminability of heterogeneous and aggregate vectors.

3.1.2. Heterogeneous versus aggregate feature vectors

A heterogeneous vector has both key hold and key interval latencies as features. We consider an aggregate vector with key press latencies as features. Let d be the number of characters in the reference phrase. With d characters, there will be $(2d-1)$ features in the heterogeneous vector and $(d-1)$ features in the aggregate vector.

Let $X_i = (x_{i1}, \ldots, x_{id})$, $i = 1, \ldots, n$ denote a key hold vector belonging to ω_1 and $X'_j = \left(x'_{j1}, \ldots, x'_{jd}\right)$, $j = 1, \ldots, n$ denote a key hold vector belonging to ω_2. Let $Y_i = (y_{i1}, \ldots, y_{id})$ denote a key interval vector belonging to ω_1 and $Y'_j = \left(y'_{j1}, \ldots, y'_{j(d-1)}\right)$, $j = 1, \ldots, n$ denote a key interval vector belonging to ω_2. The feature vectors $(X_i, Y_i) = (x_{i1}, \ldots, x_{id}, y_{i1}, \ldots, y_{i(d-1)})$ and $\left(X'_j, Y'_j\right) = \left(x'_{j1}, \ldots, x'_{jd}, y'_{j1}, \ldots, y'_{j(d-1)}\right)$ represent heterogeneous vectors of ω_1 and ω_2 respectively. The feature vectors $Z_i = (x_{i1} + y_{i1}, \ldots, x_{i(d-1)} + y_{i(d-1)})$ and $Z'_j = \left(x'_{j1} + y'_{j1}, \ldots, x'_{j(d-1)} + y'_{j(d-1)}\right)$ represent aggregate vectors of ω_1 and ω_2, respectively.

To simplify our analysis, for all samples of ω_1 and ω_2, we assume (as done in Jain and Waller (1978)) that the difference between key hold latencies and the difference between key interval latencies is the same, i.e., $x_{i1} - x'_{j1} = x_{i2} - x'_{j2} = \cdots = x_{id} - x'_{jd} = r$ and $y_{i1} - y'_{j1} = y_{i2} - y'_{j2} = \cdots = y_{id} - y'_{jd} = r$, $i = 1, \ldots, n$ and $j = 1 \ldots, n$.

Assuming $\Sigma = A$, the Mahalanobis distance (3) between the classes ω_1 and ω_2 with heterogeneous vectors (X_i, Y_i), $i = 1, \ldots, n$ and $\left(X'_j, Y'_j\right)$, $j = 1, \ldots, n$ is

$$R_{A,2d-1} = \left(\frac{1}{8nh_1h_2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}[\alpha_{2d-1}(2d-1)r^2 + 2\beta_{2d-1}r^2(2d-1)(d-1)]$$
$$= K[\alpha_{2d-1}(2d-1) + 2\beta_{2d-1}(2d-1)(d-1)] \tag{5}$$

and the Mahalanobis distance between ω_1 and ω_2 with aggregate feature vectors Z_i, $i = 1, \ldots, n$ and Z'_j, (X'_j, Y'_j) is

$$R_{A,d-1} = K[\alpha_{d-1}(d-1) + \beta_{d-1}(d-1)(d-2)]. \tag{6}$$

The difference between the discriminability of heterogeneous and aggregate vectors is

$$\Delta_1 = R_{A,2d-1} - R_{A,d-1}. \tag{7}$$

Fig. 3(a) shows how Δ_1 varies with the number of characters when the correlation between features is small. For example, when $\rho = 0.02$, Δ_1 increases steadily until about 30 characters and then gradually decreases thereafter; indicating that for short reference phrases, heterogeneous features offer more discrimination than aggregate features. However, as the number of characters in the reference phrase increase, the difference in the discriminability of heterogeneous and aggregate vectors decreases. Fig. 3(b) shows how Δ_1 varies with d when there are stronger correlations between features. Unlike the gradual raise and decrease observed in Fig. 3(a), Δ_1 in Fig. 3(b) increases more steeply within 10 characters and decreases thereafter. As the ρ value increases from 0.1 to 0.4, Δ_1 decreases more rapidly as the number of characters increase.

Assuming $\Sigma = B$, the Mahalanobis distance (4) between ω_1 and ω_2 with heterogeneous vectors (X_i, Y_i), $i = 1, \ldots, n$, and (X'_j, Y'_j), $j = 1, \ldots, n$ is

$$R_{B,2d-1} = \left(\frac{1}{8nh_1h_2}\right)\sum_{i=1}^{n}\sum_{j=1}^{n}[2\alpha r^2 + \gamma(2d-3)r^2 + 4\beta(d-1)r^2]$$
$$= K[2\alpha + \gamma(2d-3) + 4\beta(d-1)] \tag{8}$$

and the Mahalanobis distance between ω_1 and ω_2 with aggregate vectors Z_i, $i = 1, \ldots, n$ and Z'_j, $j = 1, \ldots, n$ is

$$R_{B,d-1} = K[2\alpha + \gamma(d-3) + \beta(d-1)]. \tag{9}$$

The difference between the discriminability of heterogeneous and aggregate vectors is

$$\Delta_2 = R_{B,2d-1} - R_{B,d-1}. \tag{10}$$

When $\Sigma = B$, Fig. 4 shows that Δ_2 increases as the number of characters in the reference phrase increase. The increase in Δ_2 is more pronounced for small correlation values (i.e., when ρ is between 0.1 and 0.3). As ρ increases, Δ_2 becomes flatter, indicating that the difference between heterogeneous and aggregate features reduces as the correlation between features increases.

In summary, our theoretical analysis shows that heterogeneous vector (with key press and key interval latencies as features) has higher discriminability than aggregate vector (with key press latencies as features). Under the assumption that features are equally correlated, heterogeneous vector has higher discriminability than aggregate vectors for short (5–30 character) reference texts when small ρ values (0.03–1.0) are assumed. For larger ρ values (0.15–0.4), the difference in discriminability rapidly decreases as the length of the reference text increases. Under the assumption that features of adjacent characters in the reference text are more correlated than features of non-adjacent characters, the difference

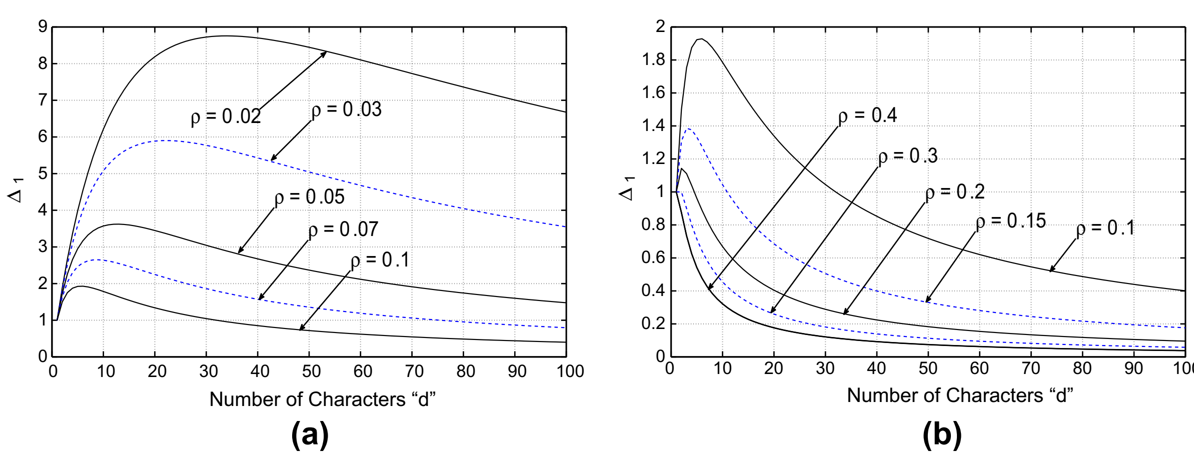

Fig. 3. Number of characters "d" in the reference text versus the difference ("Δ_1") between the discriminability of heterogeneous and aggregate vectors – (a) when ρ (correlation) varies from 0.02 to 0.1 and (b) when ρ varies from 0.1 to 0.4. The number of characters varies from 1 to 100.

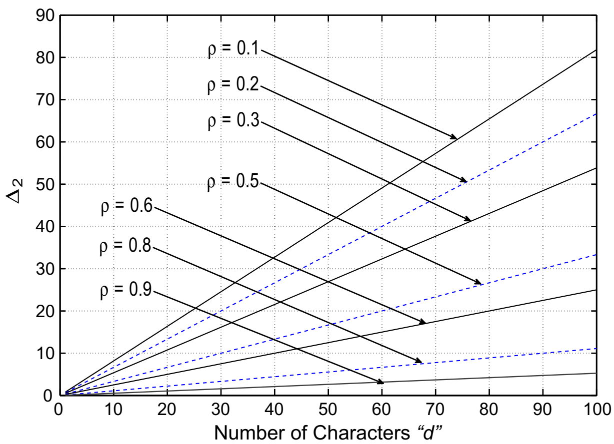

Fig. 4. Number of characters "d" in the reference text versus the difference ("Δ_2") in the discriminability of heterogeneous and aggregate vectors. The number of characters varies from 1 to 100 characters. The correlation "ρ" varies from 0.1 to 0.9.

in discriminability increases with the length of the reference text, albeit modestly for higher ρ values.

Next, we present experiments and discuss results.

4. Experiments and results

4.1. Dataset

The dataset used in the experiments was collected at Louisiana Tech University's computer science laboratory during November–December 2002 and was used in our earlier work (Sheng et al., 2005). Here, we briefly discuss the dataset. For a detailed description, see Sheng et al. (2005).

Keystroke patterns were collected using a program developed in Microsoft Foundation Classes. Forty-three users participated in the data collection process. We used "**master of science in computer science**" as the reference text. We used this particular sentence because: (1) it was easy to remember and simple to type for most users, given that the 43 users were mainly graduate students in Computer Science and (2) we wanted a sentence that was neither too short nor too long. If the sentence is too short, we would have only a few features to work with. If the sentence is too long, enrollment becomes time consuming and tiresome for the users.

Previous studies (e.g., Joyce and Gupta, 1990) have shown that familiarity with the text helps produce consistent typing patterns. To familiarize with the text before enrollment, we encouraged the users to practice typing the text as many times as they wished. Once the users became comfortable with the text, they were asked to type the text nine times for enrollment.

In a real-world scenario, users access computers at their own will and convenience. To simulate this, authentication samples were collected at the convenience and availability of the users. Users were allowed to provide as many samples as they wanted over a period of three weeks. Unfortunately, out of the 43 enrolled, 10 never returned to provide authentication samples. For authentication, we collected 873 samples from 33 users with each user providing between 6 and 102 samples. The average number of authentication samples per user was 26.45 with 21.006 standard deviation.

During enrollment and authentication, the program forced users to retype the reference text from the beginning if they made a typo or pressed DELETE or BACKSPACE keys. Typing the reference text yielded 32 key hold latencies and 26 key interval latencies. All latencies involving the SPACE key were ignored.

4.2. Classifiers for user recognition

We used four classifiers for user recognition: naive Bayes (Mitchell, 1997), tree augmented naive Bayes (Friedman et al., 1997), k-nearest neighbor (Duda et al., 2000), and ridge logistic regression (leCessie and vanHouwelingen, 1992). We choose these classifiers because: they are among the most popular in the classification literature; are known to address both linear (naive Bayes and ridge logistic regression) and non-linear (tree augmented naive Bayes and k-nearest neighbor) classification problems; and were readily available in WEKA machine learning software (Witten and Frank, 2005). The classifiers were trained on the nine enrollment samples provided by each of the 33 users. The 873 authentication samples were used as the test set. For naive Bayes and tree augmented naive Bayes classifiers, features were discretized using the method presented in (Fayyad and Irani, 1993).

4.3. User recognition results

We present user recognition results with homogeneous, heterogeneous, and aggregate feature vectors and their combinations. In our experiments, we considered eight feature vectors (see Table 2). They are: (1) two homogeneous vectors, one vector with key hold (KH) latencies and another with key interval (KI) latencies; (2) two aggregate vectors, one vector with key press (KP) latencies and another with trigraphs; (3) a heterogeneous vector with KH and KI latencies; (4) two combination vectors, one vector with KH and KP latencies and another with KI and KP latencies; and (5) a combination vector with KH, KI, and KP latencies.

In Table 2, we show the user recognition accuracies of naive Bayes and tree augmented naive Bayes classifiers. Both classifiers achieve the highest accuracy with the combination feature vector containing KH, KI, and KP latencies. The heterogeneous feature vector (i.e., [KH,KP]) achieves the second highest accuracy. The difference in accuracies of the winning feature vector and the heterogeneous vector is 1.72% with naive Bayes and 2.3% with tree augmented naive Bayes. Heterogeneous and combination (i.e., [KH,KP], [KI,KP], and [KH,KI,KP]) vectors clearly outperform homogeneous and aggregate vectors by considerable margins. The difference between accuracies of the heterogeneous vector and the aggregate vector with KP latencies is 19.244% with naive Bayes and 18.0985% with tree augmented naive Bayes. Trigraphs have the lowest recognition accuracies.

In Fig. 5, we plot the differences in the recognition accuracies when heterogeneous and aggregate (i.e., KP) vectors are used with naive Bayes and tree augmented naive Bayes classifiers. The recog-

Table 2
Recognition accuracies of naive Bayes (NB) and tree augmented naive Bayes (TAN) classifiers with homogeneous, heterogeneous, and aggregate feature vectors and their combinations.

Feature vector	Features	Percentage accuracy (rank)	
		NB	TAN
Homogeneous	KH	77.205 (6)	72.394 (7)
	KI	82.5888 (5)	83.2761 (5)
Aggregate	KP	74.9141 (7)	75.4868 (6)
	Trigraph	42.9553 (8)	44.559 (8)
Heterogeneous	KH, KI	94.1581 (2)	93.5853 (2)
Homogeneous and aggregate	KH, KP	92.3253 (3)	92.4399 (3)
	KI, KP	88.3162 (4)	88.5452 (4)
Heterogeneous and aggregate	KH, KI, KP	**95.8763 (1)**	**95.8763 (1)**

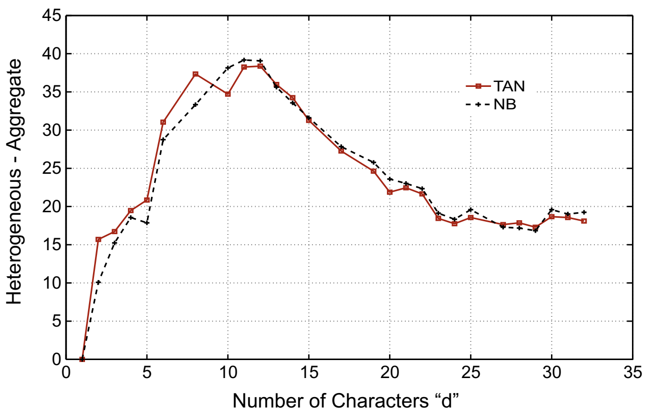

Fig. 5. Difference in the recognition accuracies of heterogeneous (key hold and key interval) and aggregate (key press) vectors with NB and TAN classifiers as the number of characters in the reference text varies from 1 to 32.

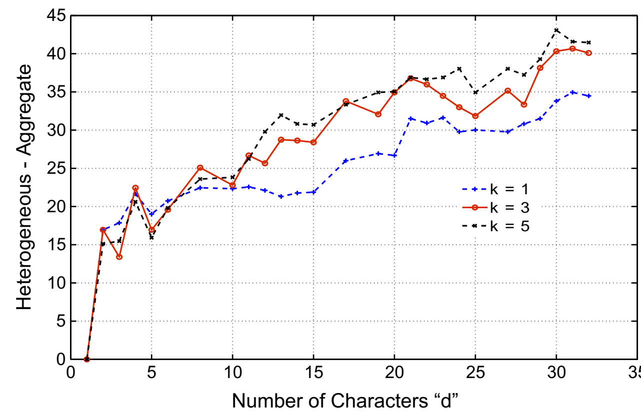

Fig. 6. Difference in the recognition accuracies of heterogeneous (key hold and key interval) and aggregate (key press) vectors with *k*-nearest neighbor classifier as the number of characters in the reference text increases from 1 to 32.

nition accuracies were calculated by increasing the length of the reference phrase, one character at a time. For example, the keystroke features of "**m**" were used to calculate the recognition accuracies at $d = 1$; the features of "**ma**" were used to calculate the accuracies at $d = 2$; the features of "**mas**" were used to calculate the accuracies at $d = 3$; and so on until $d = 32$. Fig. 5 shows that the differences in accuracy peaks at 12 characters and decreases gradually thereafter.

In Table 3, we show the user recognition accuracies of *k*-nearest neighbor classifier, when *k* is set to 1, 3, and 5. When *k* is 3 and 5, the *k*-nearest neighbor classifier achieves highest recognition accuracy with the heterogeneous ([KH, KI]) vector. When $k = 1$, the [KH, KP] combination vector has the highest recognition accuracy, followed by the heterogeneous vector with 0.0873% accuracy difference. With *k*-nearest neighbor classifier, the [KI, KP] combination feature vector shows a lackluster performance, performing worse than the homogeneous vectors. In contrast, the heterogeneous vector outperforms homogeneous and aggregate vectors by considerable margins. The difference in accuracies of the heterogeneous vector and the aggregate (KP) vector is 34.4788% when $k = 1$, 40.0917% when $k = 3$, and 41.4662% when $k = 5$. The aggregate (KP and trigraph) vectors have the lowest recognition accuracies.

In Fig. 6, we plot the differences in the recognition accuracies when heterogeneous ([KH, KI]) and aggregate (KP) vectors are used with *k*-nearest neighbor classifier. The plots in Fig. 6 show that the difference in recognition accuracies increase with the number of characters in the text. The "peaking" behavior observed with the other classifiers, (see Figs. 5 and 7), did not occur with the *k*-nearest neighbor classifier. A closer inspection of the data leading to the plots revealed that, as the number of characters increased, the

Table 3
Recognition accuracies of the *k*-nearest neighbor classifier with homogeneous, heterogeneous, and aggregate features and their combinations.

Feature vector	Features	Percentage accuracy (rank)		
		$k = 1$	$k = 3$	$k = 5$
Homogeneous	KH	82.5888 (4)	83.2761 (4)	79.6105 (4)
	KI	64.6048 (5)	57.6174 (6)	58.3047 (5)
Aggregate	KP	51.7755 (7)	46.9645 (7)	45.5899 (7)
	Trigraph	36.4261 (8)	36.3116 (8)	35.9679 (8)
Heterogeneous	KH, KI	86.2543 (2)	**87.0561 (1)**	**87.0561(1)**
Homogeneous & Aggregate	KH, KP	**86.9416 (1)**	85.567 (3)	84.6506 (3)
	KI, KP	55.6701 (6)	55.5556 (5)	54.6392 (6)
Heterogeneous & Aggregate	KH, KI, KP	84.8797 (3)	86.5979 (2)	85.1088 (2)

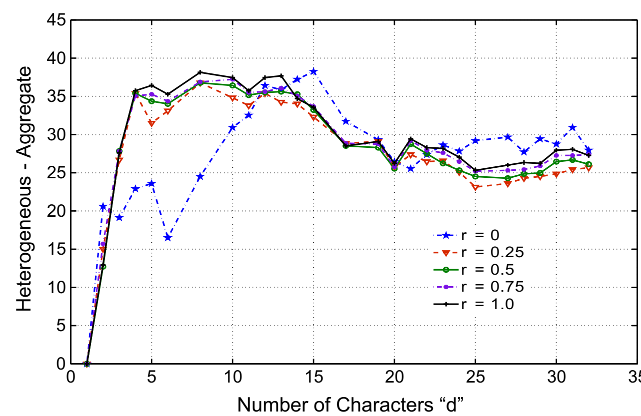

Fig. 7. Difference in the recognition accuracies of heterogeneous (key hold and key interval) and aggregate (key press) vectors with ridge logistic regression classifier.

accuracies of *k*-nearest neighbor with the KP vector increased at a much slower rate compared to the accuracies with heterogeneous vector. For example, as the number of characters increased from 17 to 32, the accuracy of *k*-nearest neighbor ($k = 1$) with KP vector changed from 51.55% to 51.76%; while with heterogeneous vector, it changed from 76.63% to 86.25%. Therefore, as the number of characters increased, *k*-nearest neighbor classifier had higher accuracies with heterogeneous vector compared to the aggregate (KP) vector, and the difference between them consistently increased.

In Table 4, we show the user recognition accuracies of ridge logistic regression classifier as the ridge parameter "*r*" varies from 0 to 1. The heterogeneous vector achieves the highest recognition accuracy when $r = \{0.25, 0.5, 0.75, 1.0\}$ and achieves second highest accuracy when $r = 0$. The combination vector [KH, KP] has the second best overall performance. The difference in accuracies of the heterogeneous feature vector and the aggregate (KP) vector ranges from 25.6586%, when $r = 0.25$, to 27.9496%, when $r = 0$. Trigraphs have the lowest recognition accuracies.

In Fig. 7, we plot the differences in the recognition accuracies when heterogeneous and aggregate (KP) vectors are used with ridge logistic regression classifier. The plots in Fig. 7 show that when *r* is 0.25, 0.5, 0.75, and 1, the difference in accuracies peaks between 8 and 10 characters and then begins to decrease as the number of characters in the reference text increase. When *r* is 0, the difference in accuracies peaks at 15 characters and then decreases gradually as the number of characters increase.

Table 4
Recognition accuracies of the ridge logistic regression classifier with homogeneous, heterogeneous, and aggregate features and their combinations. The ridge parameter is denoted as *r*. The ranks are same when *r* = 0.25, *r* = 0.5, *r* = 0.75, and *r* = 1.

Feature vector	Feature	Percentage Accuracy (Rank)				
		$r = 0$	$r = 0.25$	$r = 0.5$	$r = 0.75$	$r = 1$
Homogeneous	KH	77.4341 (4)	87.0561 (4)	87.1707	87.5143	88.6598
	KI	69.4158 (5)	72.6231 (5)	72.9668	73.5395	73.8832
Aggregate	KP	60.252 (7)	67.354 (7)	66.7812	65.7503	66.2085
	Trigraph	30.9278 (8)	41.3517 (8)	39.5189	39.1753	37.8007
Heterogeneous	KH, KI	88.2016 (2)	**93.0126 (1)**	**92.8981**	**93.1271**	**93.4708**
Homogeneous and aggregate	KH, KP	**88.6598 (1)**	91.5235 (2)	91.7526	91.9817	91.9817
	KI, KP	69.3013 (6)	73.3104 (6)	73.0813	73.1959	73.0813
Heterogeneous and aggregate	KH, KI, KP	76.0596 (3)	89.3471 (3)	89.5762	89.8053	89.5762

In summary, the results show that the heterogeneous feature vector (i.e., vector of key hold and key interval latencies) *outperforms* aggregate vectors (i.e., vectors with key press latencies or trigraphs as features). The *heterogeneous vector consistently ranked highest or second highest*, while aggregate vectors performed worst with all classifiers. Figs. 5–7 show that the difference between heterogeneous and aggregate (key press) vectors peaked between 8 and 15 characters with naive Bayes, tree augmented naive Bayes, and ridge logistic regression; and peaked at 30 characters with *k*-nearest neighbor classifier. This strongly suggests that for short reference texts, a heterogeneous vector offers higher discriminability than an aggregate vector. The heterogeneous vector consistently outperformed [KH,KP] combination feature vector. However, overall the [KH,KP] combination vector performed better than [KI,KP] and [KH,KI,KP] combination vectors. Though [KI,KP] and [KH,KI,KP] vectors performed well with naive Bayes and tree augmented naive Bayes classifiers, they showed suboptimal performance with k-nearest neighbor and ridge logistic regression classifiers, suggesting that the *classifiers have varying degrees of sensitivity to aggregation of features.*

4.4. Feature selection analysis

In feature selection analysis, we compare the discriminability of heterogeneous and aggregate feature vectors with feature subsets obtained using three filter based feature selection methods. The methods are: ReliefF, correlation based feature selection (CFS), and consistency based feature selection (CNS). We choose these methods for the following reasons: (1) the methods are state-of-the-art and are often used in machine learning applications; (2) each method selects features quite differently (i.e., ReliefF is an instance based method, CFS uses a correlation criterion, and CNS uses a consistency criterion); and (3) several studies (e.g., Hall et al., 2003; Yang and Pedersen, 1997) have demonstrated the superior performance of these methods. We briefly discuss the three feature selection methods.

ReliefF (Kononenko, 1994) considers a feature as discriminative if its values are as close as possible for instances of the same class and are as different as possible for instances of different classes. In each iteration, ReliefF selects an instance randomly and finds its nearest neighbors from each class in the dataset. The values of the features of the nearest neighbors are compared with the selected instance and the difference between the values is used to update relevance scores for each feature. After a predefined number of iterations, features with the highest relevance scores are selected.

Correlation based feature selection (Hall, 2000) method selects features that have the maximum correlation (or dependence) with the class and simultaneously, have the minimum correlation with already selected features. Several types of correlation based feature selection methods have been proposed in the literature (e.g., Kwak and Choi, 2002; Battiti, 1994). We use the CFS method of Hall (2000).

Consistency based feature selection method searches for a combination of features whose values partition the data into subsets containing a single class majority (i.e., the method searches for feature subsets with strong class purity). Entropy, inconsistency measure (Liu and Setiono, 1996), and information gain can be used to measure the consistency of a feature subset. In our experiments, we use the consistency measure used in (Hall et al., 2003).

Different search strategies (e.g., exhaustive, forward selection, backward elimination, floating forward selection, and floating backward elimination) can be used in conjunction with CFS and CNS. An exhaustive search over the feature space is the only strategy that guarantees an optimal subset. However, because of its combinatorial complexity, an exhaustive search can become impractical even for a modest number of features. Sequential forward selection and backward elimination select (eliminate) one feature at a time and are computationally very attractive strategies. However, because a feature cannot be added or removed once selected, these strategies suffer from nesting problems. Floating forward selection and floating backward elimination are similar to a sequential search, but with the property to backtrack the addition or deletion of features (and therefore, do not suffer as much from nesting problems). However, the problem with floating search is that it may not select the desired number of features.

In our feature selection analysis, we used sequential forward selection and backward elimination strategies in conjunction with CFS and CNS methods. The reason for choosing these strategies is that they ensure that the feature subsets contain the specified number of features. ReliefF ranks each individual feature and therefore is not associated with any search strategy.

4.5. Results of feature selection analysis

We used the feature selection methods to construct feature subsets from a pool of 84 features containing 32 key press, 26 key interval, and 26 key press latencies. We varied the number of features in a subset based on which vector the subset is being

Table 5
The composition of features in heterogeneous vector and in feature subsets constructed using ReliefF, CFS, and CNS methods. Each subset contains 58 features. The proportions of key hold, key interval, and key press latencies in a subset is indicated in parenthesis.

Feature vector	Key hold	Key interval	Key press
Heterogeneous	32 (55.17%)	26 (42.83%)	–
ReliefF	32 (55.17%)	14 (24.14%)	12 (20.69%)
CFS + forward selection	19 (32.76%)	17 (29.31%)	22 (37.93%)
CFS + backward elimination	24 (41.38%)	17 (29.31%)	17 (29.31%)
CNS + forward selection	31 (53.45%)	26 (42.83%)	1 (1.72%)
CNS + backward elimination	7 (12.07%)	25 (43.10%)	26 (44.83%)

Table 6
Comparison of recognition accuracies with heterogeneous vector and with feature subsets constructed by ReliefF, CFS, and CNS methods. Accuracies of feature subsets which are higher than the heterogeneous vector are underlined. NN denotes nearest neighbor classifier and LR denotes ridge logistic regression.

Method	Heterogeneous (KH, KI)	ReliefF	CFS (FS)	CNS (FS)	CFS (BE)	CNS (BE)
Naive Bayes	94.1581	94.1581	94.3872	93.5853	88.0871	89.2325
TAN	93.5853	94.3872	90.2635	93.8144	88.8889	89.2325
NN (k = 1)	86.2543	86.4834	82.7033	85.3379	76.6323	65.6357
NN (k = 3)	87.0561	86.9416	82.9324	87.1707	77.5487	64.7194
NN (k = 5)	87.0561	84.5361	82.1306	85.567	77.8923	66.5521
LR (r = 0.0)	88.2016	83.5052	78.3505	84.9943	77.0905	73.425
LR (r = 0.25)	93.0126	89.4616	88.4307	92.5544	86.0252	78.236
LR (r = 0.50)	92.8981	89.3471	88.0871	92.5544	86.0252	78.0069
LR (r = 0.75)	93.1271	89.3471	88.2016	92.5544	86.0252	78.5796
LR (r = 1.0)	93.4708	89.2325	88.3162	92.7835	86.0252	78.8087

compared to, i.e., we compare subsets containing 26 features with an aggregate vector containing 26 key press latencies. Similarly, we compare subsets containing 58 features with the heterogeneous vector containing 32 key hold and 26 key interval latencies.

In Table 5, we show the number and proportion of key hold, key interval, and key press latencies in each subset constructed by the feature selection methods. The feature selection methods created quite diverse subsets. ReliefF and CNS (forward selection) selected more than 75% of features from key hold and key interval latencies; CNS (backward elimination) selected about 87% of features from key interval and key press latencies; and CFS (forward and backward) selected roughly 30% of features from all three latencies.

In Table 6, we compare the accuracies of heterogeneous vector with feature vectors obtained by ReliefF, CFS, and CNS methods. In only three instances (underlined in Table 6), the accuracies of heterogeneous vector are less than the accuracies of the features selected by the feature selection methods. Even in these three instances, the difference in the accuracies is small, ranging between 0.23% and 0.80%. Overall, the accuracies in Table 6, indicate that heterogeneous vector, in most cases, is better than different equal-sized vectors obtained from the feature selection methods. In Table 7, we show the number and proportion of key hold, key interval, and key press latencies in each subset constructed by the feature selection methods. ReliefF selected all key hold latencies; CNS (forward selection) selected 23 of 26 features from key

Table 7
The composition of features in an aggregate vector and in feature subsets constructed using ReliefF, CFS, and CNS methods. Each subset contains 26 features. The proportions of key hold, key interval, and key press latencies in a subset is indicated in parenthesis.

Feature vector	Key hold	Key interval	Key press
Aggregate (key press)	–	–	26 (100%)
ReliefF	26 (100%)	–	–
CFS + forward selection	7 (26.92%)	10 (38.46%)	9 (34.61%)
CFS + backward elimination	12 (46.15%)	1 (3.85%)	13 (50.0%)
CNS + forward selection	23 (88.46%)	1 (3.85%)	2 (7.69%)
CNS + backward elimination	0	1 (3.85%)	25 (96.15%)

Table 8
Comparison of recognition accuracies with aggregate vector and with feature subsets constructed by ReliefF, CFS, and CNS methods. Accuracies of feature subsets which are higher than the aggregate vector are underlined. NN denotes nearest neighbor classifier and LR denotes ridge logistic regression.

Method	Aggregate KP	ReliefF	CFS (FS)	CNS (FS)	CFS (BE)	CNS (BE)
Naive Bayes	74.9141	73.0813	89.2325	93.6999	53.8373	74.4559
TAN	75.4868	69.3013	84.7652	93.4708	55.0974	74.4559
NN (k = 1)	51.7755	79.6105	70.2176	84.5361	54.7537	50.5155
NN (k = 3)	46.9645	84.9943	70.9049	83.8488	57.6174	45.1317
NN (k = 5)	45.5899	79.9542	72.966	83.6197	56.9301	43.8717
LR (r = 0.0)	60.252	70.4467	75.8305	76.6323	49.1409	58.4192
LR (r = 0.25)	67.354	84.4215	79.7251	86.7125	68.0412	66.7812
LR (r = 0.50)	66.7812	85.4525	80.1833	86.827	69.1867	65.9794
LR (r = 0.75)	65.7503	86.3688	80.8706	86.5979	69.7595	65.5212
LR (r = 1.0)	66.2085	86.827	80.756	86.7125	69.1867	64.7194

Table 9
Comparison of feature vector performance when short and long reference texts were used for fixed text keystroke authentication.

Text length	Paper	Feature vector performance
Short (6–32 character user IDs, passwords, and sentences)	Hosseinzadeh and Krishnan (2008)	Achieved best performance (4.3% FAR at 4.8% FRR) with key hold and key release combination vector. Achieved worst performance (21.9% FAR at 2.3% FRR) with aggregate (key press) vector
	Arajo et al. (2005)	Achieved best performance (1.89% FAR at 1.45% FRR) with key hold, key interval, and key press combination vector
	Obaidat and Sadoun (1997)	Achieved best results (0% FAR at 0%FRR) with heterogenous vector. Heterogeneous vector outperformed homogeneous vectors
	Sheng et al. (2005)	Achieved 1.44% FAR at 7.79% FRR with a heterogeneous vector. Other vectors were not tested
	Yu and Cho (2003)	Achieved best result (0% FAR at 3.54% FRR) with a heterogeneous vector selected using SVM wrapper based feature selection
Long (683 character paragraph)	Bergandano et al. (2002)	Achieved best results (0% FAR at 7.27% FRR) and (0.0028% FAR at 5.91% FRR) with aggregate (trigraph) vector

hold latencies; CNS (backward elimination) selected 25 of 26 features from key press latencies; CFS (forward selection) selected roughly 30% of its features from all three latencies; and CFS (backward elimination) selected almost 50% of its features from key hold latencies and another 50% from key press latencies.

In Table 8, we compare the accuracies of aggregate (key press) vector with vectors obtained by ReliefF, CFS, and CNS methods. The features selected by CFS (forward) and CNS (forward) outperform the aggregate vector by considerable margins. The features selected by ReliefF, except with naive Bayes and tree augmented naive Bayes classifiers, outperforms the aggregate vector. Also, the features selected with CFS (backward) perform better than the aggregate vector with nearest neighbor and ridge logistic regression classifier. Overall, the comparison demonstrates that discriminability can be considerably improved by choosing different combinations of features (e.g., 7 key hold, 10 key interval, and 9 key press latencies chosen by CFS forward selection) instead of using key press latencies alone as features.

5. Conclusions

We revisit the question raised in the Introduction – which feature vector, heterogeneous, aggregate, or a combination of both, is more discriminative? The answer is that a heterogeneous vector has higher discriminability than an aggregate vector, especially when the reference text is short. However, as the length of the text increases, the difference in the discriminability between heterogeneous and aggregate vectors tends to decrease. We observed this phenomenon in our theoretical analysis (Fig. 3), when we assumed equal correlation among features, and also in our user recognition results (Figs. 5 and 7).

Recognition accuracies reported in Tables 2–4 show that [KH, KP] and [KH, KI, KP] combination vectors consistently outperformed aggregate vectors. Because the dimensionality of [KH, KP] and [KH, KI, KP] vectors is considerably higher than the aggregate vectors, (i.e., 26 features in key press vector compared to 56 in [KH, KP] and 84 in [KH, KI, KP]), one can argue that the higher dimensionality alone has contributed to the superior performance of combination vectors. However, the results of feature selection analysis dismiss this argument. Table 8 shows that the aggregate vector is consistently outperformed by different combinations of "equal-sized" feature subsets (containing 26 features), suggesting that aggregation inherently causes loss in discriminability. On the other hand, results from our user recognition and feature selection experiments show that heterogeneous vector performs better than combination vectors in most cases.

Our results are consistent with the results reported in the fixed text authentication literature. In several studies, which used short reference texts, the best performance was achieved with heterogeneous or combination vectors. On the other hand, Bergandano et al. (2002) used a lengthy reference text (683 character paragraph) and reported good results with an aggregate (trigraph) vector. In Table 9, we summarize the feature vector performance of some fixed text authentication studies. However, it should be noted that the comparison of feature vectors based on the results reported in the literature is more suggestive than conclusive. The problem with relying on these results alone is that there are too many differences in the evaluation conditions (e.g., size of reference phrases, training sample sizes, number of testing attempts, updating training samples, etc.) that make comparisons across studies impractical.

In this study, we addressed a fundamental question concerning the discriminability of heterogeneous and aggregate keystroke vectors, from three complementary perspectives – intuitive reasoning, theoretical analysis, and empirical validation. We opine that the insights gained from our work will benefit practitioners and researchers of keystroke dynamics, by indicating when it is appropriate to use heterogeneous and aggregate vectors for fixed text keystroke authentication.

Acknowledgments

This work is supported in part by the Louisiana BoR Grant No. LEQSF (2007-12)-ENH-PKSFI-PRS-03 and the USAF Contract No. FA8750-09-C-0191.

References

Arajo, L.C.F., Sucupira, L.H.R., Lizrraga, M.G., Ling, L.L., Yabu-Uti, J.B.T., 2005. User authentication through typing biometrics features. IEEE Trans. Signal Process. 53 (2), 851–855.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Trans. Neural Networks 5 (4), 537–550.

Bergandano, F., Gunetti, D., Picardi, C., 2002. User authentication through keystroke dynamics. ACM Trans. Inform. System Secur. 5 (4), 362–397.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification, second ed. Wiley-Interscience Publication, New York.

Fayyad, U.M., Irani, K., 1993. Multiinterval discretization of continuous-valued attributes for classification learning. In: Pro. Thirteenth Internat. Joint Conf. on Artificial Intelligence, Chambery, France, pp. 1022–1027.

Filho, J.R.M., Freire, E.O., 2006. On the equalization of keystroke timing histograms. Pattern Recognition Lett. 27 (13), 1440–1446.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Machine Learn. 29 (2–3), 131–163.

Giot, R., El-Abed, M., Rosenberger, C., September 2009. Keystroke dynamics with low constraints svm based passphrase enrollment. In: IEEE 3rd Internat. Conf. on Biometrics: Theory, Applications, and Systems, Washington, DC, pp. 1–6.

Gray, R.M., 2006. Toeplitz and Circulant Matrices: A review (Foundations and Trends in Communications and Information), first ed. Now Publishers Inc., Hanover, MA.

Haider, S., Abbas, A., Zaidi, A.K., December 2000. A multi-technique approach for user identification through keystroke dynamics. In: IEEE Internat. Conf. on Systems, Man, and Cybernetics, Nashville, TN, pp. 1336–1341.

Hall, M., September 2000. Correlation-based feature selection for discrete and numeric class machine learning. In: Proc. 17th Internat. Conf. on Machine Learning, pp. 359–366.

Hall, M.A., Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans. Knowl. Data Eng. 15 (6), 1437–1447.

Heydorn, R., 1967. An upper bound estimate on classification error. IEEE Trans. Inform. Theory IT-14, 783–784.

Hosseinzadeh, D., Krishnan, S., 2008. Gaussian mixture modeling of keystroke patterns for biometric applications. IEEE Trans. Systems Man Cybernet. 38 (6), 816–826.

Jain, A., Waller, W., 1978. On the optimal number of features in the classification of multivariate Gaussian data. Pattern Recognition 29 (13), 1873–1876.

Joyce, R., Gupta, G., 1990. Identity authentication based on keystroke latencies. Comm. ACM 33 (2), 168–176.

Kailath, T., 1967. The divergence and bhattacharyya distance measures in signal selection. IEEE Trans. Comm. Technology COM-15, 52–60.

Killourhy, K.S., Maxion, R.A., September 2009. Comparing anomaly-detection algorithms for keystroke dynamics. In: IEEE/IFIP Internat. Conf. on Dependable Systems and Networks, Lisbon, Portugal, pp. 125–134.

Kononenko, I., September 1994. Estimating attributes: analysis and extensions of relief. In: Proc. Seventh European Conf. on Machine Learning, pp. 171–182.

Kwak, N., Choi, C.-H., 2002. Input feature selection by mutual information based on parzen window. IEEE Trans. Pattern Anal. Machine Intell. 24 (12), 1667–1671.

Lainiotis, D.G., 1969. A class of upper bounds on probability of error for multihypotheses pattern recognition. IEEE Trans. Inform. Theory IT-15, 730–731.

le Cessie, S., van Houwelingen, J.C., 1992. Ridge estimators in logistic regression. Appl. Statist. 41 (1), 191–201.

Liu, H., Setiono, R., 1996. A probabilistic approach to feature selection – a filter solution. In: Proc. 13th Internat. Conf. Machine Learning, pp. 319–327.

Mitchell, T., 1997. Machine Learning, first ed. McGraw Hill, New York.

Obaidat, M.S., Sadoun, B., 1997. Verification of computer users using keystroke dynamics. IEEE Trans. Systems Man Cybernet – Part B 27 (2), 261–269.

Parzen, E., 1962. On estimation of a probability density function and mode. Ann. Math. Statist. 33, 1065–1076.

Revett, K., de Magalhaes, S.T., Santos, H., December 2005. Data mining a keystroke dynamics based biometrics database using rough sets. In: Portuguese Conf. on Artificial Intelligence, Washington, DC, pp. 188–191.

Robinson, J.A., Liang, V.M., Chambers, J.A.M., MacKenzie, C.L., 1998. Computer user verification using login string keystroke dynamics. IEEE Trans. Systems Man Cybernet. – Part A 28 (2), 236–241.

Sheng, Y., Phoha, V.V., Rovnyak, S.M., 2005. A parallel decision tree-based method for user authentication based on keystroke patterns. IEEE Trans. Systems Man Cybernet. – Part B: Cybernet. 35 (4), 826–833.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, second ed. Morgan Kaufmann, San Francisco.
Yang, Y., Pedersen, J.O., September 1997. A comparative study on feature selection in text categorization. In: Proc. Fourteenth Internat. Conf. on Machine Learning, San Francisco, CA, USA, pp. 412–420.
Yu, E., Cho, S., 2003. Novelty detection approach for keystroke dynamics identity verification. In: Proc. Fourth Internat. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL 2003), pp. 1016–1023.