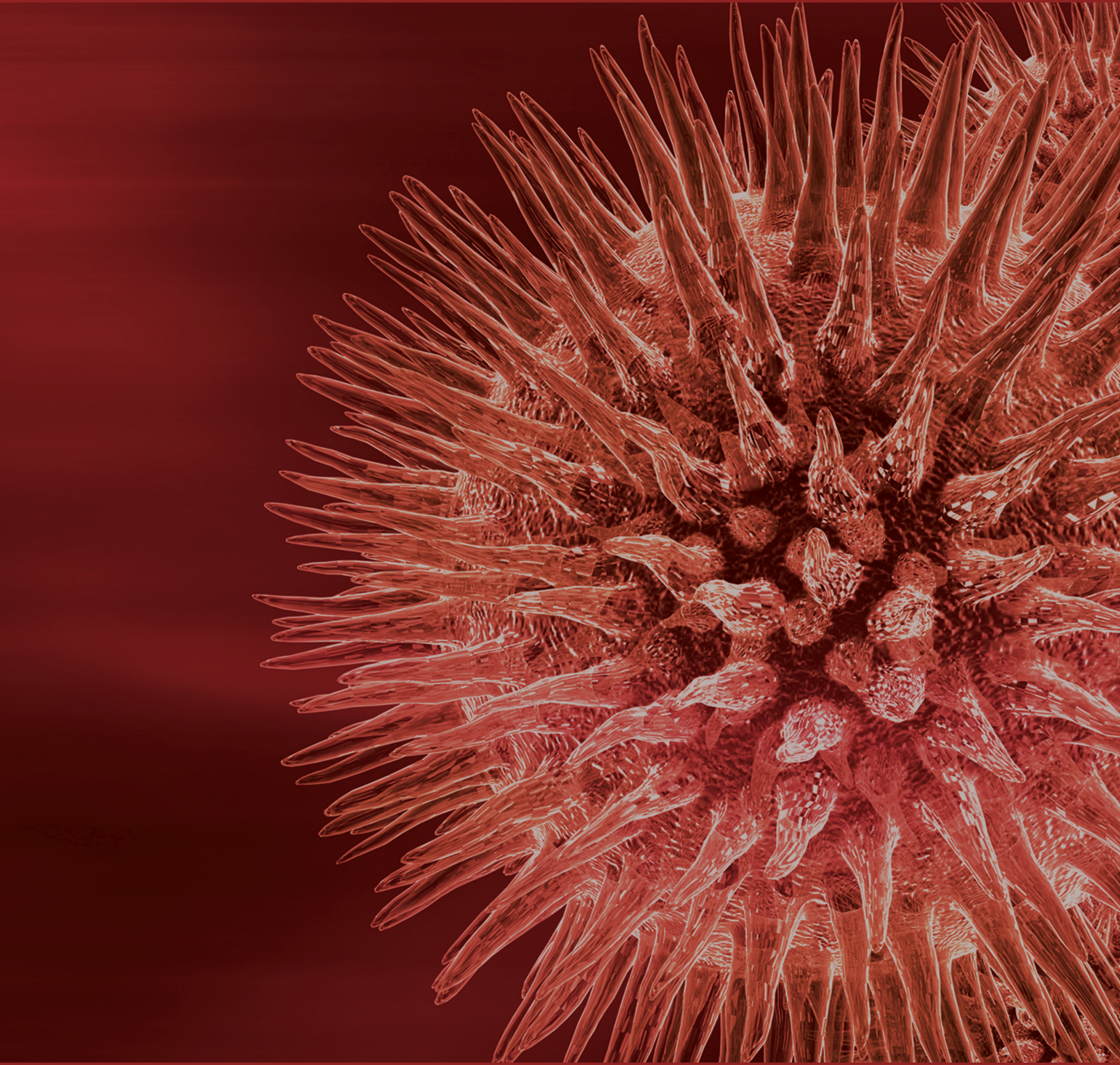
BioMed Research International

Biometrics and Biosecurity 2013

Guest Editors: Tai-hoon Kim, Sabah Mohammed, and Wai-Chi Fang



Biometrics and Biosecurity 2013

BioMed Research International

Biometrics and Biosecurity 2013

Guest Editors: Tai-hoon Kim, Sabah Mohammed, and Wai-Chi Fang

Contents

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 734642, 2 pages
http://dx.doi.org/10.1155/2013/734642

Editorial

Biometrics and Biosecurity 2013

Tai-hoon Kim,[1] Sabah Mohammed,[2] and Wai-Chi Fang[3]

[1] *Department of Convergence Security, Sungshin Women's University, Dongseon-dong-3-ga, Seongbuk-gu, Seoul, Republic of Korea*
[2] *Department of Computer Science, Lakehead University, 955 Oliver Road, Thunder Bay, ON, Canada P7B 5E1*
[3] *National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan*

Correspondence should be addressed to Tai-hoon Kim; taihoonn@daum.net

Received 17 November 2013; Accepted 17 November 2013

We are very happy to publish this special issue. This issue contains 11 articles that come from various countries, among which we mention Saudi Arabia, Republic of Korea, Macau, Canada, Australia, China, the Czech Republic, India, and Japan.

Biometrics and Biosecurity 2013 focused on the various aspects of advances in biometrics and biosecurity. This special issue will provide a chance for academic and industry professionals to discuss recent progress, problems, and solutions in the area of biometrics and its application, biosecurity measures, and biosafety protocols, including development, implementation, strategies, and policies.

In "*An improved biometrics-based remote user authentication scheme with user anonymity*," the authors proposed an improved scheme eradicating the flaws of the previous scheme. The proposed scheme not only withstands security problems found in the previous scheme but also provides some extra features with the mere addition of only two hash operations.

In the paper "*The quantitative overhead analysis for effective task migration in biosensor networks*," authors presented a quantitative overhead analysis for effective task migration in biosensor networks. A biosensor network is the key technology which can automatically provide accurate and specific parameters of a human in real time. The results of performance evaluation showed that task execution time is greatly influenced by a cluster ratio and different processing time of biosensor nodes.

The objective of the paper "*Evaluation of stream mining classifiers for real-time clinical decision support system: a case study of blood glucose prediction in diabetes therapy*" is to find out the most suitable classifier for rt-CDSS, and therefore the authors compared them in a diabetes therapy scenario. Also the authors tested the performance of the classifier candidate all-rounded with a real-time case study, as a preliminary step to validate the efficacy of the rt-CDSS as a whole.

In "*Classifying human voices by using hybrid SFX time-series preprocessing and ensemble feature selection*," the authors focused on the comparison of effects of various popular data mining algorithms on multiple datasets. The authors' experiment consisted of classification tests over four typical categories of human voice data, namely, female and male, emotional speech, speaker identification, and language recognition.

In the paper "*Secure encapsulation and publication of biological services in the cloud computing environment*," secure encapsulation and publication for bioinformatics software products based on web service were presented and the basic function of biological information was realized in the cloud computing environment. In the encapsulation phase, the workflow and function of the bioinformatics software were conducted, the encapsulation interfaces were designed, and the runtime interaction between users and computers was simulated. In the publication phase, the execution and management mechanisms and principles of the GRAM components were analyzed.

In "*Image analysis of endoscopic ultrasonography in submucosal tumor using fuzzy inference*," the authors proposed a method to extract areas of GIST and lipoma automatically from the standardized ultrasonic image to assist those endoscopists. The authors also proposed an algorithm to differentiate GIST from non-GIST by fuzzy inference from such images after applying an ROC curve with mean and standard deviation of the brightness information.

In "*A study on user authentication methodology using numeric password and fingerprint biometric information*," user authentication was performed that uses biometric information and passwords of users. The user cannot change user's fingerprint information, but the user has set a password to change easily. So this authentication system provides security and flexibility. Because it makes a password key that utilizes the user's fingerprint and numeric password, an attacker does not take advantage of leaked passwords.

This article "*New optical methods for liveness detection on fingers*" was devoted to new optical methods, which are supposed to be used for liveness detection on fingers. First the authors described basics about fake finger use in the fingerprint recognition process and possibilities of liveness detection. Then the authors continued with introduction of three new liveness detection methods, which the authors developed and tested in the scope of the authors' research activities—the first one was based on measurement of pulse, the second one was based on variations of optical characteristics caused by pressure change, and the last one was based on reaction of skin to illumination with different wavelengths.

In the paper "*Designing a bioEngine for detection and analysis of base string on an affected sequence in high-concentration regions*," authors designed an algorithm for the bioengine. Searching for homologues had become a routine operation of biological sequences in 4 × 4 combination with a different subsequence (word size). This program takes advantage of the high degree of homology between such sequences to construct an alignment of the matching regions.

In "*Statistical fractal models based on GND-PCA and its application on classification of liver diseases*," a new method was proposed to establish the statistical fractal model for liver diseases classification. Firstly, the fractal theory was used to construct the high-order tensor, and then Generalized N-dimensional principal component analysis (GND-PCA) was used to establish the statistical fractal model and select the feature from the region of the liver; at the same time different features had different weights. Finally, the support vector machine optimized ant colony (ACO-SVM) algorithm was used to establish the classifier for the recognition of the liver disease.

The paper "*HyDEn: a hybrid steganocryptographic approach for data encryption using randomized error-correcting DNA codes*" presented a novel hybrid DNA Encryption (HyDEn) approach that uses randomized assignments of unique error-correcting DNA Hamming code words for single characters in the extended ASCII set. HyDEn relied on custom-built quaternary codes and a private key used in the randomized assignment of code words and the cyclic permutations applied to the encoded message.

Acknowledgment

Achieving such a high quality of papers would have been impossible without the huge work that was undertaken by the editorial board members and external reviewers. We take this opportunity to thank them for their great support and cooperation.

Tai-hoon Kim
Sabah Mohammed
Wai-Chi Fang

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 491289, 9 pages
http://dx.doi.org/10.1155/2013/491289

Research Article

An Improved Biometrics-Based Remote User Authentication Scheme with User Anonymity

Muhammad Khurram Khan[1] **and Saru Kumari**[2]

[1] *King Saud University, P.O. Box 92144, Riyadh 11653, Saudi Arabia*
[2] *Department of Mathematics, Agra College, Agra, Dr. B. R. A. University, Agra, Uttar Pradesh 282002, India*

Correspondence should be addressed to Muhammad Khurram Khan; mkhurram@ksu.edu.sa

Received 4 August 2013; Accepted 2 September 2013

Academic Editor: Sabah Mohammed

The authors review the biometrics-based user authentication scheme proposed by An in 2012. The authors show that there exist loopholes in the scheme which are detrimental for its security. Therefore the authors propose an improved scheme eradicating the flaws of An's scheme. Then a detailed security analysis of the proposed scheme is presented followed by its efficiency comparison. The proposed scheme not only withstands security problems found in An's scheme but also provides some extra features with mere addition of only two hash operations. The proposed scheme allows user to freely change his password and also provides user anonymity with untraceability.

1. Introduction

In the last two decades, digital authentication has originated as a preferred method to authenticate remote users over insecure networks. After the first proposal of user authentication scheme by Lamport [1], considerable amount of research has been conducted in this field of which schemes [1–25] are few examples. In due course of time user authentication schemes underwent many changes. Initial schemes were based only on password [1–4], then schemes were based on smart card and password [5–13], and reliability of biometrics authentication over traditional password-based authentication gave rise to biometrics-based user authentication schemes [14–20].

In 2010, Li and Hwang [19] proposed a biometrics-based user authentication scheme. In 2011, Das [26] examined Li-Hwang's scheme and observed problems in login and authentication phase, in password change phase, and in biometrics verification mechanism of the scheme. Das depicted that user's smart card does not validate the inputted password during login phase which leads to useless computations in login and authentication phase. Owing to the same reason, Das further showed that the scheme suffers from incorrect password updating problem. Thus, Das proposed an improvement [26] of Li-Hwang's scheme and claimed their scheme to be free from problems observed in Li-Hwang's scheme. According to Das, their scheme [26] also provides mutual authentication. In 2012, An [27] pointed out that Das's scheme [26] deviates from the author's claim since an adversary can mount impersonation attacks and password guessing attack once he gets a chance to extract values from the smart card of the legal user. Thereby An [27] proposed an enhanced scheme to eradicate the flaws of Das's scheme.

In this paper, we review An's biometrics-based user authentication scheme. We show that An's scheme is vulnerable to the security problems to which Das's scheme is susceptible like online and offline password guessing attacks, user and server impersonation attacks, lack of mutual authentication, and lack of user anonymity. Besides, An's scheme lacks password change facility which is an important part of password-based user authentication schemes. We remove drawbacks from An's scheme by means of proposing an improved user authentication scheme. In addition, to resist various security threats, the proposed scheme incorporates features of password changing and user anonymity. The rest of this paper is arranged as follows. In Section 2, we review An's user authentication scheme. Section 3 is about cryptanalysis of An's scheme. In Section 4, we present our improved scheme. Section 5 is about security analysis of the improved

Table 1: Notations with their description.

Notations	Description
R	Trusted registration centre
S_i	Server
C_i	User
ID_i	Identity of C_i
PW_i	Password of C_i
B_i	Biometric template of C_i
SC_i	Smart card of C_i
K_i	Random number chosen by C_i
R_c	Random number generated by SC_i of C_i
R_s	Random number generated by S_i
U_a	Attacker
x_s and y_s	Secret keys maintained by S_i
$h(\cdot)$	One-way hash function
$\oplus$	Bitwise XOR operator
$\parallel$	Concatenation operator

scheme. In Section 6, we compare the improved scheme with related schemes. Finally, the conclusion is presented in Section 7.

2. Review of An's Scheme

The notations useful in this paper are summarized along with their description in Table 1. In this section, we review An's scheme [27] which is an enhanced version of Das's scheme [26]. It has three phases: registration phase, login phase and authentication phase. Registration phase is carried over a secure channel whereas login phase, and authentication phase are carried over an insecure channel. There are three participants in the scheme, the user (C_i), the server (S_i), and the registration centre (R), where R is assumed to be a trusted party. Details of each phase are given in the following subsections.

2.1. Registration Phase. In the beginning of scheme, the registration centre R and the user C_i carry out this phase involving the following steps.

(1) C_i submits his identity ID_i and information $(\mathrm{PW}_i \oplus K_i)$ containing password to R via a secure channel. C_i also submits information $(B_i \oplus K_i)$ containing his biometrics via the specific device to R; here K_i is a random number chosen by C_i.

(2) Rcomputes $f_i = h(B_i \oplus K_i)$, $r_i = h(\mathrm{PW}_i \oplus K_i) \oplus f_i$, and $e_i = h(\mathrm{ID}_i \parallel x_s) \oplus r_i$, where x_s is a secret key generated and maintained by S_i. Then R stores $\{\mathrm{ID}_i, f_i, e_i, h(\cdot)\}$ in a smart card SC_i for user and provides it to C_i via a secure channel.

(3) On receiving $\mathrm{SC}_i = \{\mathrm{ID}_i, f_i, e_i, h(\cdot)\}$, the user stores the random number K_i into SC_i issued by R so that now $\mathrm{SC}_i = \{\mathrm{ID}_i, f_i, e_i, h(\cdot)\}$.

2.2. Login Phase. When the user C_i wishes to login the server S_i, the user and his smart card SC_i perform the following steps.

(1) C_i inserts his smart card into a card reader and inputs his biometrics information B_i on the specific device. SC_i computes $h(B_i \oplus K_i)$ and verifies if $f_i = h(B_i \oplus K_i)$ or not. If this biometrics information matches, C_i passes the biometrics verification.

(2) C_i inputs his ID_i and PW_i; then SC_i generates a random number R_c and computes the following equations:

$$\begin{gathered} r_i' = h(\mathrm{PW}_i \oplus K_i) \oplus f_i, \\ M_1 = e_i \oplus r_i', \\ M_2 = M_1 \oplus R_c, \\ M_3 = h(M_1 \parallel R_c). \end{gathered} \tag{1}$$

(3) C_i sends the login request $= \{\mathrm{ID}_i, M_2, M_3\}$ to S_i.

2.3. Authentication Phase. On receiving the request login $= \{\mathrm{ID}_i, M_2, M_3\}$ from C_i, the server S_i and the user C_i perform the following steps to authenticate each other.

(1) S_i first checks the format of ID_i. If ID_i is valid, S_i computes $M_4 = h(\mathrm{ID}_i \parallel x_s)$ and $M_5 = M_2 \oplus M_4$.

(2) S_i checks if $M_3 = h(M_4 \parallel M_5)$ or not. If both are equal, it generates a random number R_s and computes the following equations:

$$\begin{gathered} M_6 = M_4 \oplus R_s, \\ M_7 = h(M_4 \parallel R_s). \end{gathered} \tag{2}$$

Then, S_i sends the reply message $= \{M_6, M_7\}$ for its authentication to C_i.

(3) On receiving $\{M_6, M_7\}$ from S_i, the user C_i computes $M_8 = M_6 \oplus M_1$ and checks if $M_7 = h(M_1 \parallel M_8)$ or not. If both are equal, C_i computes $M_9 = h(M_1 \| R_c \| M_8)$ and sends the reply message $\{M_9\}$ for its authentication to S_i.

(4) On receiving $\{M_9\}$ from C_i, the server checks if $M_9 = h(M_4 \| M_5 \| R_s)$ or not. If both are equal, S_i accepts the login request $= \{\mathrm{ID}_i, M_2, M_3\}$ of C_i.

3. Cryptanalysis of An's Scheme

This section is about security problems in An's scheme. Here we show that an attacker U_a can mount different types of attacks on the scheme. Independent researches by Kocher and Messerges [28, 29] show that it is possible to extract the values stored inside a smart card. So we assume that U_a can extract out parameters stored inside a user's smart card.

3.1. Online Password Guessing Attack. If U_a obtains the smart card SC_i of user C_i and extracts [28, 29] the values $\{\mathrm{ID}_i, f_i, e_i, K_i, h(\cdot)\}$ stored inside it, then he can mount online password guessing attack as explained below.

(1) U_a computes

$$\begin{aligned} e_i \oplus f_i &= [h(\mathrm{ID}_i \parallel x_s) \oplus r_i] \oplus f_i \\ &= [h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i) \oplus f_i] \oplus f_i \\ &= [h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i)] \end{aligned} \tag{3}$$

to obtain $[h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i)]$.

(2) U_a guesses PW_a as user's possible password and computes $M_{1a} = [e_i \oplus f_i] \oplus h(\mathrm{PW}_a \oplus K_i)$. Then U_a computes $M_{2a} = M_{1a} \oplus R_{ca}$ and $M_{3a} = h(M_{1a} \parallel R_{ca})$, where R_{ca} is the random number generated by the system of U_a. *He* sends $\{\mathrm{ID}_i, M_{2a}, M_{3a}\}$ as login request to S_i.

(3) If U_a does not receive any response from S_i then he repeats step (2) with some other guess for user's password. But if U_a receives response message from S_i, then it implies that his guessed password PW_a is correct.

3.2. Offline Password Guessing Attack. In the scheme, U_a can easily identify the login request corresponding to a smart card since both contain the identity of user. If U_a extracts [28, 29] the values $\{\mathrm{ID}_i, f_i, e_i, K_i, h(\cdot)\}$ from the smart card SC_i of user C_i and intercepts the login request $= \{\mathrm{ID}_i, M_2, M_3\}$ from open network, then he can mount offline password guessing attack as explained below.

(1) U_a computes

$$\begin{aligned} e_i \oplus f_i &= [h(\mathrm{ID}_i \parallel x_s) \oplus r_i] \oplus f_i \\ &= [h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i) \oplus f_i] \oplus f_i \\ &= [h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i)] \end{aligned} \tag{4}$$

to obtain $[h(\mathrm{ID}_i \parallel x_s) \oplus h(\mathrm{PW}_i \oplus K_i)]$.

(2) U_a guesses PW_a as user's possible password and computes $M_{1a} = [e_i \oplus f_i] \oplus h(\mathrm{PW}_a \oplus K_i)$.

(3) U_a computes $R_{ca} = M_2 \oplus M_{1a}$ and $M_{3a} = h(M_{1a} \parallel R_{ca})$, and finally compares M_{3a} with M_3. For $M_{3a} \neq M_3$, he repeats from step (2) with some other guess for user's password. But if $M_{3a} = M_3$, then it provides U_a with the exact password PW_i of C_i.

3.3. User Impersonation Attack. As just discussed in previous subsections, U_a can guess a user's password if he obtains the smart card of user. It is noticeable that the successful process of password guessing (online or offline manner) also yields $M_{1a} = h(\mathrm{ID}_i \parallel x_s)$. In fact, $h(\mathrm{ID}_i \parallel x_s)$ is the key value required to compute a valid login request or valid reply messages. Further, U_a has easy access to user's identity ID_i from $\mathrm{SC}_i = \{\mathrm{ID}_i, f_i, e_i, K_i, h(\cdot)\}$ or from the login request $= \{\mathrm{ID}_i, M_2, M_3\}$ of C_i. Having $h(\mathrm{ID}_i \parallel x_s)$ and ID_i in hand, U_a can impersonate the user C_i as explained below.

(1) U_a generates a random number R_{ca} in his system and computes

$$\begin{aligned} M_{2a} &= M_{1a} \oplus R_{ca}, \\ M_{3a} &= h(M_{1a} \parallel R_{ca}). \end{aligned} \tag{5}$$

Then U_a sends the login request $= \{\mathrm{ID}_i, M_{2a}, M_{3a}\}$ to S_i.

(2) On receiving $\{\mathrm{ID}_i, M_{2a}, M_{3a}\}$, the server S_i first checks the format of ID_i. Clearly, S_i would proceed further because ID_i is the identity of a legitimate registered user and hence it is in valid format.

(3) S_i computes $M_4 = h(\mathrm{ID}_i \parallel x_s)$ and $M_5 = M_{2a} \oplus M_4$ and checks if $M_{3a} = h(M_4 \parallel M_5)$; clearly it would hold. Therefore S_i believes that the login request $= \{\mathrm{ID}_i, M_{2a}, M_{3a}\}$ is from the legitimate user.

(4) S_i generates a random number R_s and computes $M_6 = M_4 \oplus R_s$ and $M_7 = h(M_4 \parallel R_s)$. Then S_i transmits the reply message $\{M_6, M_7\}$.

(5) On receiving $\{M_6, M_7\}$ from S_i, the attacker U_a first obtains the random number R_s by computing $M_{8a} = M_6 \oplus M_{1a}$. Next, it computes $M_{9a} = h(M_{1a} \| R_{ca} \| M_8)$ and sends $\{M_{9a}\}$ to S_i.

(6) On receiving $\{M_9\}$, the server S_i checks if $M_9 = h(M_4 \| M_5 \| R_s)$ or not. Clearly, this would hold, so S_i will accept the login request $= \{\mathrm{ID}_i, M_{2a}, M_{3a}\}$.

3.4. Server Impersonation Attack. U_a can easily impersonate the legal server S_i to cheat the user C_i whose information $\{\mathrm{ID}_i$ and $M_{1a} = h(\mathrm{ID}_i \parallel x_s)\}$ he possesses as described in Section 3.3. To masquerade as S_i the attacker proceeds in the following manner.

(1) U_a can easily recognize the login request $= \{\mathrm{ID}_i, M_2, M_3\}$ of C_i transmitted over open channel as he possesses the identity ID_i of C_i. So when C_i sends his login request $= \{\mathrm{ID}_i, M_2, M_3\}$ to S_i, the attacker U_a intercepts and blocks it from reaching S_i.

(2) U_a first obtains the random number R_c by computing $M_{5a} = M_2 \oplus M_{1a}$. Next, he generates a random number R_{sa} in his system and computes $M_{6a} = M_{1a} \oplus R_{sa}$ and $M_{7a} = h(M_{1a} \parallel R_{sa})$. Then U_a transmits the reply message $\{M_{6a}, M_{7a}\}$ to C_i.

(3) On receiving $\{M_{6a}, M_{7a}\}$, the user C_i first obtains the random number R_{sa} by computing $M_8 = M_{6a} \oplus M_1$, where $M_1 = h(\mathrm{ID}_i \parallel x_s)$. Next, he checks if $M_{7a} = h(M_1 \parallel M_8)$ or not. Clearly, this equivalence will hold and hence C_i will believe that he is communicating with the intended server. However, it is the clever attacker U_a who is deceiving C_i.

3.5. Lack of Mutual Authentication. Like Das's scheme [26], the enhanced scheme by An also fails to resist user impersonation attack and server impersonation attack as described in Sections 3.3 and 3.4. In fact, if U_a extracts values $\{\mathrm{ID}_i, f_i, e_i, K_i, h(\cdot)\}$ from the smart card SC_i of user C_i and successfully obtains the secret value $h(\mathrm{ID}_i \parallel x_s)$, then he can easily craft valid login request and reply messages so as to deceive the legal user or the legal server. Therefore, the scheme loses mutual authentication feature.

3.6. Lack of User Anonymity. In An's scheme, C_i sends $\{\mathrm{ID}_i, M_2, M_3\}$ as his login request to S_i through an insecure channel. User's identity ID_i is openly available if an attacker U_a intercepts the login request of C_i from the open channel. Moreover, identity ID_i is also stored inside user's smart card SC_i. Having ID_i in hand, it is easy for U_a to craft threats against C_i. To the worst, U_a may be able to compromise user's biometrics information which would result in serious consequences. Thus, the scheme does not provide user anonymity.

4. The Proposed Scheme

In this section, we propose a new user authentication scheme which is an improvement of An's scheme. In addition to resist the security problems found in An's scheme, it also provides password change phase with which user can change his password at his will. It has four phases: registration phase, login phase, authentication phase and password change phase. Registration phase, and password change phase are carried over a secure channel whereas login phase and authentication phase are carried over an insecure channel. It also consists of three participants, the user (C_i), the server (S_i), and the registration centre (R). In the proposed scheme, the server maintains two secret keys x_s and y_s. Details of each phase along with Figure 1 are given in the following.

4.1. Registration Phase. Before starting the scheme, the registration centre R and the user C_i carry out this phase involving the following steps.

(1) C_i submits his identity ID_i and information $(\mathrm{PW}_i \oplus K_i)$ containing password to R via a secure channel. C_i also submits information $(B_i \oplus K_i)$ containing his biometrics via a specific device to R; here K_i is a random number chosen by C_i.

(2) R computes the following values:

$$\begin{aligned} f_i &= h(B_i \oplus K_i), \\ r_i &= h(\mathrm{PW}_i \oplus K_i) \oplus f_i, \\ c_i &= h(x_s \parallel y_s) \oplus f_i, \\ e_i &= h(\mathrm{ID}_i \parallel x_s) \oplus r_i, \end{aligned} \tag{6}$$

where R stores $\{c_i, e_i, h(\cdot)\}$ in a smart card SC_i for user. Then R provides $\mathrm{SC}_i = \{c_i, e_i, h(\cdot)\}$ and f_i to the user C_i via a secure channel.

(3) On receiving $[\mathrm{SC}_i = \{c_i, e_i, h(\cdot)\}\ \&\ f_i]$, the user computes the following values:

$$\begin{aligned} g_i &= (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus f_i, \\ j_i &= (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus K_i, \end{aligned} \tag{7}$$

where C_i inserts g_i and j_i into SC_i issued by R so that now $\mathrm{SC}_i = \{c_i, e_i, g_i, j_i, h(\cdot)\}$.

4.2. Login Phase. When the user C_i wishes to login the server S_i, the user and his smart card SC_i perform the following steps.

(1) C_i inserts his smart card into a card reader, keys in his identity ID_i, and password PW_i and inputs his biometrics information B_i on the specific device.

(2) SC_i retrieves $f_i \leftarrow (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus g_i$ and $K_i \leftarrow (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus j_i$. It then checks if $f_i = h(B_i \oplus K_i)$ or not. If this biometrics information matches, C_i passes the biometrics verification; otherwise SC_i terminates the sesion. This process also verifies the correctness of inserted ID_i and PW_i.

(3) SC_i generates a random number R_c and computes the following equations:

$$\begin{gathered} r_i = h(\mathrm{PW}_i \oplus K_i) \oplus f_i, \\ M_1 = c_i \oplus f_i \quad (\text{which is indeed } h(x_s \parallel y_s)), \\ M_2 = e_i \oplus r_i \quad (\text{which is indeed } h(\mathrm{ID}_i \parallel x_s)), \\ M_3 = M_1 \oplus R_c \quad (\text{which is indeed } h(x_s \parallel y_s) \oplus R_c), \\ M_4 = (M_1 \parallel R_c) \oplus \mathrm{ID}_i \\ (\text{which is indeed } [(h(x_s \parallel y_s) \parallel R_c) \oplus \mathrm{ID}_i]), \\ M_5 = h(M_2 \parallel R_c), \\ (\text{which is indeed } h(h(\mathrm{ID}_i \parallel x_s) \parallel R_c)). \end{gathered} \tag{8}$$

(4) C_i sends the login request = $\{M_3, M_4, M_5\}$ to S_i.

4.3. Authentication Phase. On receiving the request login = $\{M_3, M_4, M_5\}$ from C_i, the server S_i and the user C_i perform the following steps to authenticate each other.

(1) S_i computes the following values:

$$\begin{gathered} M_6 = h(x_s \parallel y_s), \\ M_7 = M_3 \oplus M_6 \quad (\text{which is indeed } R_c), \\ \mathrm{ID}_i = M_4 \oplus (M_6 \parallel M_7). \end{gathered} \tag{9}$$

(2) S_i checks the format of ID_i. If ID_i is valid, S_i computes $M_8 = h(\mathrm{ID}_i \parallel x_s)$. It then checks if $M_5 = h(M_8 \parallel M_7)$.

User (C_i)		Registration centre (R)
Registration phase		Computes $f_i = h(B_i \oplus K_i)$,
Chooses $\mathrm{ID_i}$, PW_i & K_i	$\xrightarrow{\{\mathrm{ID}_i, (\mathrm{PW}_i \oplus K_i), (B_i \oplus K)\}}$	$r_i = h(\mathrm{PW}_i \oplus K_i) \oplus f_i$, $c_i = h(x_s \Vert y_s) \oplus f_i$, and
	$\xleftarrow{[\mathrm{SC}_i = \{c_i, e_i, h(\cdot)\} \,\&\, f_i]}$	$e_i = h(\mathrm{ID}_i \Vert x_s) \oplus r_i$
Computes		
$g_i = (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus f_i$ and		
$j_i = (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus K_i$		
Insert g_i & j_i into SC_i so that $\mathrm{SC}_i = \{c_i, e_i, g_i, j_i, h(\cdot)\}$		
User (C_i)		Server (S_i)
Login and authentication phase		
U: inserts ID_i, PW_i & B_i		Computes $M_6 = h(x_s \Vert y_s)$,
SC : $f_i \leftarrow (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus g_i$ and		$M_7 = M_3 \oplus M_6$, and
$K_i \leftarrow (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus j_i$		$\mathrm{ID}_i = M_4 \oplus (M_6 \Vert M_7)$
For $f_i = h(B_i \oplus K_i)$ computes		For correct ID_i format computes
$r_i = h(\mathrm{PW}_i \oplus K_i) \oplus f_i$,	$\xrightarrow{\{M_3, M_4, M_5\}}$	$M_8 = h(\mathrm{ID}_i \Vert x_s)$
$M_1 = c_i \oplus f_i, M_2 = e_i \oplus r_i$,		For $M_5 = h(M_8 \Vert M_7)$ computes
$M_3 = M_1 \oplus R_c, M_4 = (M_1 \Vert R_c) \oplus \mathrm{ID}_i$,		$M_9 = M_8 \oplus R_s$ and
$M_5 = h(M_2 \Vert R_c)$.		$M_{10} = h(M_8 \Vert R_s)$
	$\xleftarrow{\{M_9, M_{10}\}}$	
Computes $M_{11} = M_9 \oplus M_2$		
For $M_{10} = h(M_2 \Vert M_{11})$		
Computes $M_{12} = h(M_2 \Vert R_c \Vert M_{11})$		
	$\xrightarrow{\{M_{12}\}}$	For $M_{12} = h(M_8 \Vert M_7 \Vert R_s)$
		Accepts login request
User (C_i)		Smart card (SC_i)
Password change phase:		$f_i \leftarrow (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus g_i$ and
U: inserts ID_i, PW_i & B_i	$\xrightarrow{\{\mathrm{ID}_i, \mathrm{PW}_i \,\&\, B_i\}}$	$K_i \leftarrow (\mathrm{ID}_i \Vert \mathrm{PW}_i) \oplus j_i$
		For $f_i = h(B_i \oplus K_i)$ asks for new password
	$\xrightarrow{(\mathrm{PW}_i)_{\mathrm{new}}}$	Computes $(g_i)_{\mathrm{new}} = (\mathrm{ID}_i \Vert (\mathrm{PW}_i)_{\mathrm{new}}) \oplus f_i$,
		$(j_i)_{\mathrm{new}} = (\mathrm{ID}_i \Vert (\mathrm{PW}_i)_{\mathrm{new}} \oplus K_i$, and
		$(e_i)_{\mathrm{new}} =$ $e_i \oplus h(\mathrm{PW}_i \oplus K_i) \oplus h((\mathrm{PW}_i)_{\mathrm{new}} \oplus K_i)$.
		$(e_i)_{\mathrm{new}} \leftarrow e_i, (g_i)_{\mathrm{new}} \leftarrow g_i, (j_i)_{\mathrm{new}} \leftarrow j_i$

Figure 1: The proposed scheme.

If both are equal, S_i generates a random number R_s and computes:

$$\begin{aligned} M_9 &= M_8 \oplus R_s \\ &\left(\text{which is indeed } h\left(\text{ID}_i \parallel x_s\right) \oplus R_s\right) \\ M_{10} &= h\left(M_8 \parallel R_s\right) \\ &\left(\text{which is indeed } h\left(h\left(\text{ID}_i \parallel x_s\right) \parallel R_s\right)\right). \end{aligned} \tag{10}$$

Then, S_i sends the reply message = $\{M_9, M_{10}\}$ for its authentication to C_i.

(3) On receiving $\{M_9, M_{10}\}$ from S_i, the user C_i computes $M_{11} = M_9 \oplus M_2$ (which is indeed R_s). It then checks if $M_{10} = h(M_2 \parallel M_{11})$ or not. If both are equal, C_i computes $M_{12} = h(M_2 \parallel R_c \parallel M_{11})$ (which is indeed $h[h(\text{ID}_i \parallel x_s) \parallel R_c \parallel R_s]$). Then C_i sends the reply message $\{M_{12}\}$ for its authentication to S_i.

(4) On receiving $\{M_{12}\}$ from C_i, the server checks if $M_{12} = h(M_8 \parallel M_7 \parallel R_s)$ or not. If both are equal, S_i accepts the login request = $\{M_3, M_4, M_5\}$ of C_i.

4.4. Password Change Phase. When the user wishes to change his old password PW_i, he invokes this phase. Details of the steps required to update the smart card SC_i with new password $(\text{PW}_i)_{\text{new}}$ are as follows.

(1) C_i inserts his smart card into a card reader, keys in his identity ID_i, and password PW_i and inputs his biometrics information B_i on the specific device.

(2) SC_i retrieves $f_i \leftarrow (\text{ID}_i \parallel \text{PW}_i) \oplus g_i$ and $K_i \leftarrow (\text{ID}_i \parallel \text{PW}_i) \oplus j_i$. It then checks if $f_i = h(B_i \oplus K_i)$ or not. If this biometrics information matches, C_i passes the biometrics verification, otherwise terminates the session. This process also verifies the correctness of inserted ID_i and PW_i. Then SC_i allows the user to enter the new password $(\text{PW}_i)_{\text{new}}$.

(3) SC_i computes the following equations:

$$\begin{aligned} \left(g_i\right)_{\text{new}} &= \left(\text{ID}_i \parallel \left(\text{PW}_i\right)_{\text{new}}\right) \oplus f_i, \\ \left(j_i\right)_{\text{new}} &= \left(\text{ID}_i \parallel \left(\text{PW}_i\right)_{\text{new}}\right) \oplus K_i, \\ \left(e_i\right)_{\text{new}} &= e_i \oplus h\left(\text{PW}_i \oplus K_i\right) \oplus h\left(\left(\text{PW}_i\right)_{\text{new}} \oplus K_i\right). \end{aligned} \tag{11}$$

(4) SC_i replaces e_i, g_i, and j_i with $(e_i)_{\text{new}}$, $(g_i)_{\text{new}}$ and $(j_i)_{\text{new}}$, respectively.

5. Security Analysis of the Proposed Scheme

In this section, we analyze security of the proposed scheme. We show that the scheme remains unaffected even if an attacker U_a extracts [28, 29] all the values stored inside a user's smart card.

5.1. Online Password Guessing Attack. On having access to user's smart card SC_i an attacker U_a can extract [28, 29] all values $\{c_i, e_i, g_i, j_i, h(\cdot)\}$ from it. In order to compute $e_i \oplus f_i$ and obtain $[h(\text{ID}_i \parallel x_s) \oplus h(\text{PW}_i \oplus K_i)]$, he requires f_i. But U_a cannot obtain f_i from $g_i = (\text{ID}_i \parallel \text{PW}_i) \oplus f_i$ as he does not know about user's identity ID_i and password PW_i. The attacker U_a can obtain $f_i \oplus K_i$ by performing $g_i \oplus j_i = [(\text{ID}_i \parallel \text{PW}_i) \oplus f_i] \oplus [(\text{ID}_i \parallel \text{PW}_i) \oplus K_i]$. Next, he can compute

$$\begin{aligned} e_i \oplus \left(f_i \oplus K_i\right) &= \left[h\left(\text{ID}_i \parallel x_s\right) \oplus r_i\right] \oplus \left(f_i \oplus K_i\right) \\ &= \left[h\left(\text{ID}_i \parallel x_s\right) \oplus h\left(\text{PW}_i \oplus K_i\right) \oplus f_i\right] \oplus \left(f_i \oplus K_i\right) \\ &= h\left(\text{ID}_i \parallel x_s\right) \oplus h\left(\text{PW}_i \oplus K_i\right) \oplus K_i. \end{aligned} \tag{12}$$

But U_a cannot compute forged M_{2a} $(= h(\text{ID}_i \parallel x_s)) = [e_i \oplus f_i \oplus K_i] \oplus h(\text{PW}_a \oplus K_i)$ using a guessed password PW_a because it requires knowledge of K_i. It is troublesome for U_a to obtain K_i because K_i is not stored in plaintext inside user's smart card but is stored securely in $j_i = (\text{ID}_i \parallel \text{PW}_i) \oplus K_i$. Further U_a cannot obtain K_i from j_i without knowing ID_i and password PW_i. Besides, U_a cannot compute M_{1a} $(= h(x_s \parallel y_s)) = (c_i \oplus f_i)$ as he does not have access to f_i. Moreover, U_a does not have ID_i of C_i as ID_i is not stored in plaintext inside user's smart card. Thus, U_a cannot compute a login request $\{M_{3a}, M_{4a}, M_{5a}\}$ in a way so as to guess user's password in an online manner. Hence, the proposed scheme withstands online password guessing attack.

5.2. Offline Password Guessing Attack. Suppose U_a obtains the smart card of some user. Though U_a can intercept login message of any user from open channel, he cannot relate a user's smart card with its corresponding login request. This is due to the fact that, unlike An's scheme, in the proposed scheme user's identity in plaintext is neither stored inside user's smart card nor transmitted in login request. As a result, U_a cannot combine values extracted from a user's smart card with values of corresponding login request to guess user's password in an offline manner. If we consider the situation that U_a somehow happens to get the correct combination of user's smart card and login request, we show that still U_a cannot mount offline password guessing attack. To guess password of C_i and then verify the guess, U_a can use $M_5 = h(M_2 \parallel R_c)$ provided that he possesses the values $\{[h(\text{ID}_i \parallel x_s) \oplus h(\text{PW}_i \oplus K_i) \oplus K_i], K_i$ and $R_c\}$ in hand. As explained in Section 5.1, U_a can obtain $[h(\text{ID}_i \parallel x_s) \oplus h(\text{PW}_i \oplus K_i) \oplus K_i]$ using $\{g_i, j_i$ and $e_i\}$ extracted [28, 29] from SC_i, but he cannot obtain the random number K_i. Besides, U_a cannot obtain the random number R_c using $M_3 = M_1 \oplus R_c$ without having M_1 $(= h(x_s \parallel y_s))$ and U_a fails to obtain M_1 $(= h(x_s \parallel y_s))$ as discussed in Section 5.1. Thus an attacker U_a cannot guess user's password in an offline manner.

5.3. User Impersonation and Server Impersonation Attack. To impersonate a legal user, U_a should possess $M_1 = h(x_s \parallel y_s)$ and $M_2 = h(\text{ID}_i \parallel x_s)$; otherwise he cannot compute a valid login request $\{M_{3a}, M_{4a}, M_{5a}\}$ or a valid reply message $\{M_{12a}\}$. The value $h(\text{ID}_i \parallel x_s)$ is equally important if U_a wishes to masquerade as legal server. Unlike An's scheme, in the proposed scheme U_a is not able to obtain M_2 $(= M_8) = h(\text{ID}_i \parallel x_s)$ while making attempts of guessing user's

Table 2: Comparison of security attributes.

Security attributes	Schemes			
	Li-Hwang's [19]	Das's [26]	An's [27]	Ours
Resist online PW_i guessing attack	No	No	No	Yes
Resist offline PW_i guessing attack	No	No	No	Yes
Resist user impersonation attack	No	No	No	Yes
Resist server impersonation attack	No	No	No	Yes
Provides mutual authentication	No	No	No	Yes
Provides PW_i change facility	Yes	Yes	No	Yes
Provides user anonymity	No	No	No	Yes

password. This is due to the fact that password guessing is not feasible as explained in Sections 5.1 and 5.2. Moreover, U_a cannot obtain $M_1 = h(x_s \parallel y_s)$ (i) from $M_3 = M_1 \oplus R_c$ obtained by intercepting the login request of C_i because of not having random number R_c and (ii) from $c_i = h(x_s \parallel y_s) \oplus f_i$ extracted from user's smart card without knowing f_i. Thus, the proposed scheme resists impersonation attacks.

5.4. Supporting Mutual Authentication. The success of mutual authentication in the proposed scheme follows directly from resistance against user impersonation attack and server impersonation attack as described in Section 5.3. In fact, U_a has many hurdles before him to act as a legal user or a legal server: (i) the secret keys x_s and y_s maintained by the server are unknown for U_a and (ii) U_a has no access to the identity ID_i of user C_i. As a result, U_a cannot compute $h(x_s \parallel y_s)$ and $h(\mathrm{ID}_i \parallel x_s)$ required to mount impersonation attacks. Besides, U_a has no method to retrieve these values either from the parameters extracted out of user's smart card or from the login request or using both. Therefore, the proposed scheme provides proper mutual authentication.

5.5. Providing User Anonymity and User Untraceability. In the proposed scheme, user's plaintext identity ID_i is completely out of scene; it is neither stored in user's smart card SC_i nor sent in any of the login-authentication messages transmitted over insecure network. If U_a extracts [28, 29] the values $\{c_i, e_i, g_i, j_i, h(\cdot)\}$ from SC_i, we explain in the following that he cannot obtain ID_i of C_i. To guess ID_i from $g_i = (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus f_i$ and from $j_i = (\mathrm{ID}_i \parallel \mathrm{PW}_i) \oplus K_i$, the attacker must have the knowledge of $\{\mathrm{PW}_i, f_i\}$ and $\{\mathrm{PW}_i, K_i\}$, respectively. U_a cannot guess out ID_i from $e_i = h(\mathrm{ID}_i \parallel x_s) \oplus r_i$ without knowing r_i and x_s. If U_a intercepts a login request $\{M_3, M_4, M_5\}$ or the reply message $\{M_9, M_{10}\}/\{M_{12}\}$, he cannot guess out ID_i using $\{M_5, M_{10}, M_{12}\}$ without the knowledge of $\{x_s, R_c$ and $R_s\}$. Besides, it is not feasible for U_a to retrieve ID_i out of $\{e_i, M_5, M_{10}, M_{12}\}$ due to one-way property of hash function. Moreover, each value $\{M_3, M_4, M_5, M_9, M_{10}, M_{12}\}$ transmitted over insecure network is dynamic in nature by virtue of random numbers R_c and R_s which are different for each session. Thus, U_a can neither obtain user's identity ID_i nor can he trace the legal user by means of observing and analyzing some fixed parameter in the login request or the reply messages. Hence, the scheme provides user anonymity as well as user untraceability.

Table 3: Comparison of computational load in terms of hash functions.

Phases	Schemes			
	Li-Hwang's [19]	Das's [26]	An's [27]	Ours
Registration phase	3 $h(\cdot)$	3 $h(\cdot)$	3 $h(\cdot)$	4 $h(\cdot)$
Login phase	2 $h(\cdot)$	2 $h(\cdot)$	3 $h(\cdot)$	3 $h(\cdot)$
Authentication phase	5 $h(\cdot)$	8 $h(\cdot)$	6 $h(\cdot)$	7 $h(\cdot)$
Total	10 $h(\cdot)$	13 $h(\cdot)$	12 $h(\cdot)$	14 $h(\cdot)$

5.6. Providing Password Change Facility. In An's scheme, once user chooses his password during registration phase, it is fixed forever as user cannot change his password at his will. Probably the author might have opined that in the presence of biometrics verification procedure there is no need of password change facility. Undoubtedly, it is very difficult to forge copy or compromise biometrics, but once compromised then biometrics cannot be changed like passwords. So we opine that if password is employed in user authentication scheme then there should be the provision to facilitate the user to freely change his password. The proposed scheme provides password changing facility with which a user can freely (without interacting with server) change his old password to a new one whenever he feels to do so. Before updating stored values with the new password $(\mathrm{PW}_i)_{\mathrm{new}}$, the smart card verifies the correctness of identity ID_i old password PW_i along with verifying the biometrics information $f_i = h(B_i \oplus K_i)$. Thus the proposed scheme provides secure and easy password changing facility.

6. Comparison

In this section, we examine the proposed scheme by means of comparing its efficiency with Li-Hwang's scheme [19], Das's scheme [26], and An's scheme [27]. Table 2 displays comparison of security attributes and Table 3 displays comparison of computational load in terms of hash functions. Comparison in Table 2 shows that the proposed scheme resists various attacks possible on schemes [19, 26, 27] and provides additional feature of user anonymity with untraceability. Besides, it also restores password change facility which is provided by original versions [19, 26] but is missing in An's scheme [27]. As Table 3 shows, the proposed scheme carries only two additional hash operations over its immediate predecessor scheme [27]. The important aspect about the proposed scheme is minor increase of two hash functions in computational load to achieve higher efficiency as compared to other schemes [19, 26, 27].

7. Conclusion

This paper shows that the recently proposed biometrics-based user authentication scheme by An is susceptible to many threats. Once an attacker obtains the smart card of a legal user, he can guess user's password and impersonate the user. Further, the attacker can also cheat the user by masquerading as the legal server. Consequently, the scheme fails to provide mutual authentication. Besides, the scheme also suffers from the restriction of static password. We have proposed a new scheme based on the design of An's scheme so as to fix the problems identified in An's scheme. In the proposed scheme an attacker cannot figure out the identity of user either from the smart card or by intercepting all login-authentication messages transmitted over insecure network. Analysis and comparison show improved performance of the proposed scheme.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-288.

References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[2] N. M. Haller, "The S/KEY one-time password system," RFC1760, February 1995.

[3] G. Horng, "Password authentication without using a password table," *Information Processing Letters*, vol. 55, no. 5, pp. 247–250, 1995.

[4] J.-K. Jan and Y.-Y. Chen, "'Paramita wisdom' password authentication scheme without verification tables," *The Journal of Systems and Software*, vol. 42, no. 1, pp. 45–57, 1998.

[5] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[6] W.-C. Ku and S.-M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 204–207, 2004.

[7] C.-I. Fan, Y.-C. Chan, and Z.-K. Zhang, "Robust remote authentication scheme with smart cards," *Computers and Security*, vol. 24, no. 8, pp. 619–628, 2005.

[8] J.-Y. Liu, A.-M. Zhou, and M.-X. Gao, "A new mutual authentication scheme based on nonce and smart cards," *Computer Communications*, vol. 31, no. 10, pp. 2205–2209, 2008.

[9] M. Kumar, M. K. Gupta, and S. Kumari, "An improved efficient remote password authentication scheme with smart card over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167–177, 2011.

[10] M. Kumar, M. K. Gupta, and S. Kumari, "An improved smart card based remote user authentication scheme with session key agreement during the verification phase," *Journal of Applied Computer Science & Mathematics*, vol. 11, no. 5, pp. 38–46, 2011.

[11] S. Kumari, M. K. Gupta, and M. Kumar, "Cryptanalysis and security enhancement of Chen *et al.'s* remote user authentication scheme using smart card," *Central European Journal of Computer Science*, vol. 2, no. 1, pp. 60–75, 2012.

[12] S. Kumari, F. B. Muhaya, M. K. Khan, and R. Kumar, "Cryptanalysis of 'a robust smart-card-based remote user password authentication scheme'," in *Proceedings of the International Symposium on Biometrics and Security Technologies*, Chengdu, China, July 2013.

[13] S. Kumari and M. K. Khan, "Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme'," *International Journal of Communication Systems*, 2013.

[14] J. K. Lee, S. R. Ryu, and K. Y. Yoo, "Fingerprint-based remote user authentication scheme using smart cards," *Electronics Letters*, vol. 38, no. 12, pp. 554–555, 2002.

[15] C.-H. Lin and Y.-Y. Lai, "A flexible biometrics remote user authentication scheme," *Computer Standards and Interfaces*, vol. 27, no. 1, pp. 19–23, 2004.

[16] M. K. Khan and J. Zhang, "Improving the security of 'a flexible biometrics remote user authentication scheme'," *Computer Standards and Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

[17] M. K. Khan, J. Zhang, and X. Wang, "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," *Chaos, Solitons & Fractals*, vol. 35, no. 3, pp. 519–524, 2008.

[18] M. K. Khan, "Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world," *IETE Technical Review*, vol. 26, no. 3, pp. 191–195, 2009.

[19] C.-T. Li and M.-S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010.

[20] M. K. Khan, S. Kumari, and M. K. Gupta, "More efficient key-hash based fingerprint remote authentication scheme using mobile device," *Computing*, 2013.

[21] M. K. Khan, S.-K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme'," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[22] M. Kumar, M. K. Gupta, and S. Kumari, "Cryptanalysis of enhancements of a password authentication scheme over insecure networks," in *Proceedings of the 4th International Conference on Contemporary Computing (IC3 '11)*, vol. 168, pp. 524–532, Noida, India, 2011.

[23] M. K. Khan, S. Kumari, and M. K. Gupta, "Further cryptanalysis of 'a remote authentication scheme using mobile device'," in *Proceedings of the 4th International Conference on Computational Aspects of Social Networks (CASoN '12)*, pp. 234–237, Sao Carlos, Brazil, November 2012.

[24] S. Kumari, M. K. Gupta, M. K. Khan, and F. T. B. Muhaya, “Cryptanalysis of ‘an improved timestamp-based remote user authentication scheme’,” in *Proceedings of the International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE ’12)*, pp. 1439–1442, Chengdu, China, June 2012.

[25] S. Kumari, M. K. Khan, and R. Kumar, “Cryptanalysis and improvement of ‘a privacy enhanced scheme for telecare medical information systems’,” *Journal of Medical Systems*, vol. 37, no. 4, article 9952, 2013.

[26] A. K. Das, “Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards,” *IET Information Security*, vol. 5, no. 3, pp. 145–151, 2011.

[27] Y. An, “Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards,” *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 519723, 6 pages, 2012.

[28] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology—CRYPTO’ 99*, pp. 388–397, Springer, Berlin, Germany, 1999.

[29] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 720834, 27 pages
http://dx.doi.org/10.1155/2013/720834

Research Article

Classifying Human Voices by Using Hybrid SFX Time-Series Preprocessing and Ensemble Feature Selection

Simon Fong,[1] Kun Lan,[1] and Raymond Wong[2]

[1] *Department of Computer and Information Science, University of Macau, Macau*
[2] *School of Computer Science and Engineering, University of New South Wales, Kensington, NSW 2052, Australia*

Correspondence should be addressed to Simon Fong; ccfong@umac.mo

Received 25 June 2013; Accepted 1 August 2013

Academic Editor: Sabah Mohammed

Voice biometrics is one kind of physiological characteristics whose voice is different for each individual person. Due to this uniqueness, voice classification has found useful applications in classifying speakers' gender, mother tongue or ethnicity (accent), emotion states, identity verification, verbal command control, and so forth. In this paper, we adopt a new preprocessing method named Statistical Feature Extraction (SFX) for extracting important features in training a classification model, based on piecewise transformation treating an audio waveform as a time-series. Using SFX we can faithfully remodel statistical characteristics of the time-series; together with spectral analysis, a substantial amount of features are extracted in combination. An ensemble is utilized in selecting only the influential features to be used in classification model induction. We focus on the comparison of effects of various popular data mining algorithms on multiple datasets. Our experiment consists of classification tests over four typical categories of human voice data, namely, Female and Male, Emotional Speech, Speaker Identification, and Language Recognition. The experiments yield encouraging results supporting the fact that heuristically choosing significant features from both time and frequency domains indeed produces better performance in voice classification than traditional signal processing techniques alone, like wavelets and LPC-to-CC.

1. Introduction

Unlike fingerprints, iris, retina, and facial feature, our voice is a kind of bodily characteristics that is useful in speaker identification but it remains relatively unexplored. Compared to other bodily features, voice is dynamic and complex, in the sense that a speech can be spoken in different languages, different tones, and in different emotions. Voice biometrics plays a central role in many biometrics applications such as speaker verification, authentication, and access control management. Furthermore voice classification potentially can apply to interactive-voice-response system for detecting the moods and tones of customers, thereby guessing if the calls are of complaints or complement, for example. More examples of voice classification have been described in our previous work [1] which attempted to classify voice data by using hierarchical time-series clustering methods. The clustering method only separates voice data into distinct groups without knowing the labels of the groups. Voice classification method trains and tests voice data into classes of known labels.

Voice classification has been studied intensively in the biometrics research community using digital signal processing methods. The signatures of the voice are expressed in numeric values in the frequency domain. There lie considerable challenges in attaining high accuracy in voice classification given the dynamic nature in the speech data, not only the contents within but also the diversity of human vocals and different ways of speeches. In this paper we tackle the classification challenges by modeling human voices as time-series in the form of stochastic signals. In contrast to deterministic signals that are rigidly periodic, stochastic signals are difficult to be modeled precisely by mathematical functions due to uncertainty in the parameters of the computational equations. Time-series of voice data are nonstationary, with their statistical characteristics change over time when spoken. As far as human voice is concerned, almost all of them are stochastic and nonstationary, meaning that their statistics are time dependent or time varying.

Given such temporal data properties, human voice that is acquired continually from the time domain would be in the form of random time-series that often has a single variable (amplitude in loudness) over time. It is believed that the statistical characteristics are changing over time during a speech but they may form some specific patterns, so some inherent information can be derived from the time-series that are useful for classification. Specifically we adopt a recent preprocessing methodology, called Statistical Feature Extraction (SFX) [2], that can effectively transform a univariate time-series voice data to a multivariate data while capturing the informative characteristics of the time-series. It is known that conventional data mining models can be deployed for classifying data with only multiple attributes. Previous work by other researchers who utilized wavelet transformation essentially converted temporal data to the representation of frequency domain format. For voice classification in this paper, elements of both time domain and frequency domain are used for obtaining the statistical characteristics of the time-series, and subsequently subject to model learning for classification that can be generically implemented by most of the available classification algorithms.

Simulation experiments are carried out over four representative types of voice data or speeches being digitized for validating the efficacy of our proposed voice classification approach based on SFX and metaheuristic feature selection. This type of feature selection will find the optimal subset of features for inducing the classification model with the highest accuracy. The four types of testing data are deliberately chosen with the purpose of covering a wide range of possible voice classification applications, such as Female and Male (FM), Emotional Speech (ES), Speaker Identification (SI), and Language Recognition (LR). Given the multiattributes which are derived from the original time-series via the preprocessing step, feature selection (FS) techniques could be applied prior to training a classification model. Our results indicate that superior performance could be achieved by using SFX and FS together over the original time-series for voice classification. The improvements are consistent over the four testing datasets with respect to the major performance indicators.

The rest of the paper is structured as follows: The previous works on classifying voice data are reviewed in Section 2; specifically their time-series transformation and feature extraction techniques are highlighted. Our proposed voice classification model which converts time-series voice data to its encoded vector representation via statistical and spectral analysis is described in detail in Section 3. A set of comparative experiments is performed by using four kinds of voice datasets, and they are reported in Section 4. Results that reinforce the efficacy of our new approach are shown in Section 5. The performance evaluation is all-rounded by considering accuracy, Kappa statistic, precision, recall, F-measure, ROC area under curve, and time cost for each dataset. Section 6 concludes this research work and suggests some future works.

2. Related Work

Human voice is stochastic, nonstationary, and bounded in frequency spectrum; hence some suitable features could be quantitatively extracted from the voice data for further processing and analysis. Over the years, different attempts have been made by previous researchers who used a variety of time-series preprocessing techniques as well as the core classification algorithms for extracting acoustic features from the raw time-series data. Their performances, however, vary.

2.1. Feature Extraction on Voice Data. Some useful features selected for the targeted acoustic surveillance are [3] weighted average delta energy (Δ_E), LPC spectrum flatness (F_{LPC}), FFT spectrum flatness (F_{FFT}), zero crossing rate (R_{ZC}), harmonicity (H), mid-level crossing rate (R_{MC}), and peak and valley count rate (R_{PV}). The classifier model used by the authors is the sliding window Hidden Markov Model (HMM). They obtained an average error rate at the range of 5%–20%. Peeters discovered more detailed acoustic features for sound description [4]. These features can be roughly grouped into temporal, energy, spectral, harmonic, perceptual, and various features. The limitation is the expensive time and space costs of computation for such full kind of feature extraction.

In the research community of signal processing, the most widely used methods for voice/speech feature extraction are Linear Prediction Coding or Linear Prediction Coefficient (LPC), Cepstral Coefficient or Cepstrum Coefficient (CC), and Mel Frequency Cepstral Coefficient (MFCC). LPC consists of finding a time-based series of n-pole infinite impulse response (IIR) filters whose coefficients better adapt to the formants of a speech signal. The main idea behind LPC is that a sample of speech can be approximated as a linear combination of past speech samples [5]. The methods for calculating LPCs include covariance method, autocorrelation (Durbin) method, lattice method, inverse filter formulation, spectral estimation formulation, maximum likelihood method, and inner product method [6].

As a general practice of pattern recognition, the final predictor coefficients are never applied because of the high variance. Instead, cepstral coefficients [7] are introduced for transforming the LPC predictor coefficients to those with more robust property. Cepstral coefficients are the inverse Fourier transform representation of the log magnitude of the spectrum. The cepstral series represents a progressive approximation of the envelope of the signal [8]. MFCC offers the best performance within six coefficients (the other five coefficients are Linear Prediction Coefficient, Linear Prediction Cepstral Coefficient, Linear Frequency Cepstral Coefficient, and Reflection Coefficient) [9]. MFCC divided the speech into frames (typically 20 ms for each frame), applied Discrete Fourier Transformation over every frame, retained the logarithm of the amplitude spectrum, smoothed the spectrum, and applied Discrete Cosine Transform [10]. Several modified MFCC methods are shown having better performance in some cases. One of them is weighted MFCC. To reduce the dimensions of feature vector while still retaining the advantages of delta and double delta features, the weighted MFCC coefficients equal the sum of MFCC coefficients, p times

Delta features and, q times double Delta features, where p and q are weights in real numbers [11]. An enhanced technique for feature recognition using Improved Features for Dynamic Time Warping (DTW) was applied as a classifier; the accuracy was between 85% and 98%. Zhou et al. designed a new Kullback-Leibler distance (KLD) based weighting Perceptual Linear Prediction (PLP) algorithm for MFCC. The KLD is defined as the distance of two continuous functions; it is a measure between reality distribution p and approximating model q. The weight is the reciprocal of this distance [12]. The word error rate was below 25%.

Similar to LPC and MFCC, PLP modifies the short-term spectrum of the speech by several psychophysically based transformations. The basic steps of PLP contain spectral analysis, critical-band spectral resolution, equal-loudness preemphasis, intensity-loudness power law, autoregressive modeling, and practical considerations [13]. But PLP is vulnerable when spectral values are modified by the frequency response of the communication channel. Thus, by employing relative spectra filtering of log domain coefficients (RASTA), we make PLP more robust to these distortions [14].

Tsrrneo Nitta used multiple mapping operators to extract topological structures, hidden in time spectrum patterns. Linear algebra is the main technique. Karhunen-Loeve transformation and linear discriminant analysis were the feature extraction methods [15]. The error rate was lower than 30%. Lee et al. proposed a new feature extraction method called independent component analysis (ICA). The purpose of an ICA network is to calculate and extract independent components from speech segment by training. Meanwhile, the weight matrix holds the basic function coefficients from the speech segment. One assumption of ICA is that the observation is the linear combination of the independent components [16]. The error rate was 5% at most.

Our proposed method uses both statistical and spectral analysis for extracting all the possible features. Subsequently it selects useful features via a metaheuristic search. The qualified features are then used to reduce the vector dimensionality of training instances for building a classification model. The features from the temporal domain contain richer statistical information than only local maxima and local minima. Our method rides on the observed current trend of fusing information from both time and frequency domains. The merit is that a nonlinear relationship is represented by the spectrum of a spectrum, so only the useful features from the frequency domain in addition to other strong statistical features from the time-domain are encoded into the multidimensional vector which of course is limited in space. Besides, residual and volatility are introduced and embedded into voice classification to produce superior classification result.

2.2. Data Mining Algorithms for Voice Classification. Some recent research tapped on the power of data mining algorithms for performing voice classification in various applications. For instance, a new method is proposed by the research team of Lee et al. [17], for prescribing personalized medicine using vocal and facial features. It is a constitution diagnostic method based solely on the individual's physical characteristics, irrespective of psychological traits, characteristics of clinical medicine, and genetic factors. They used Support Vector Machine (SVM) on a software package called LIBLINEAR (L2-loss SVM dual type) for doing voice classification.

As a contribution to telemedicine in home telemonitoring, Maunder et al. [18] investigated the possibility of automatically detecting the sound signatures of activities of daily living of an elderly patient using nonintrusive and reliable methods. A Gaussian mixture model (GMM) classifier was used to differentiate sound activities. Their experiments yielded encouraging results; with recognition accuracies in the range 70% to 100% can be consistently obtained using different microphone-pair positions, under all but the most severe noise conditions.

For biomedical applications, Chenausky et al. made an important contribution in acoustic analysis of Parkinson's disease (PD) speech [19]. The speech of 10 PD patients and 12 normal controls was analyzed for syllable rate and variability, syllable length patterning, vowel fraction, voice-onset time variability, and spirantization. These were normalized by the controls' standard deviation to represent distance from normal and combined into a composite measure. A feedback device that was developed from these findings could be useful to clinicians adjusting deep brain stimulation (DBS) parameters, as a means for ensuring they do not unwittingly choose DBS settings which impair patients' communication.

In our previous work in [1], surveyed several approaches have been studied, such as Artificial Neural Networks (ANN), Support Vector Machines (SVMs), Hidden Markov Models (HMMs), and Gaussian Mixture Models (GMMs). They have been used for training up a classification model with predefined voice samples for voice recognition. A summary of the techniques by which majority of research works used was shown in [1]. In particular, an approach by using unsupervised clustering was described in [1], where priori labeled samples are not required, and the characteristic groupings will be dedicated by the samples themselves. Voiceprints who share similar features will be placed into distinctive groups that represent some labels about the speakers. Subsequently a decision tree (classifier) can be built after studying and confirming the characteristic groups.

Above all the methods a forementioned, encoding techniques from the frequency domains are used as sole features for modeling the voice samples. A single classification algorithm was used specifically for conducting the validation experiment in the literature. In this paper, we advocate combining features from both time and frequency domains, for a throughout coverage of all the voice data characteristics. Then feature selection is used to reduce the dimensionality of the training samples. This way, a minimum subset of relevant features is ensured, and they could be applied into most types of classification models without any limit of a specific type.

3. Proposed Method in Constructing a Voice Classification Model

The SFX preprocessing methodology that is adopted in our research is efficient. Its main merit lies in its ability to transform voice data from one-dimensional to multidimensional

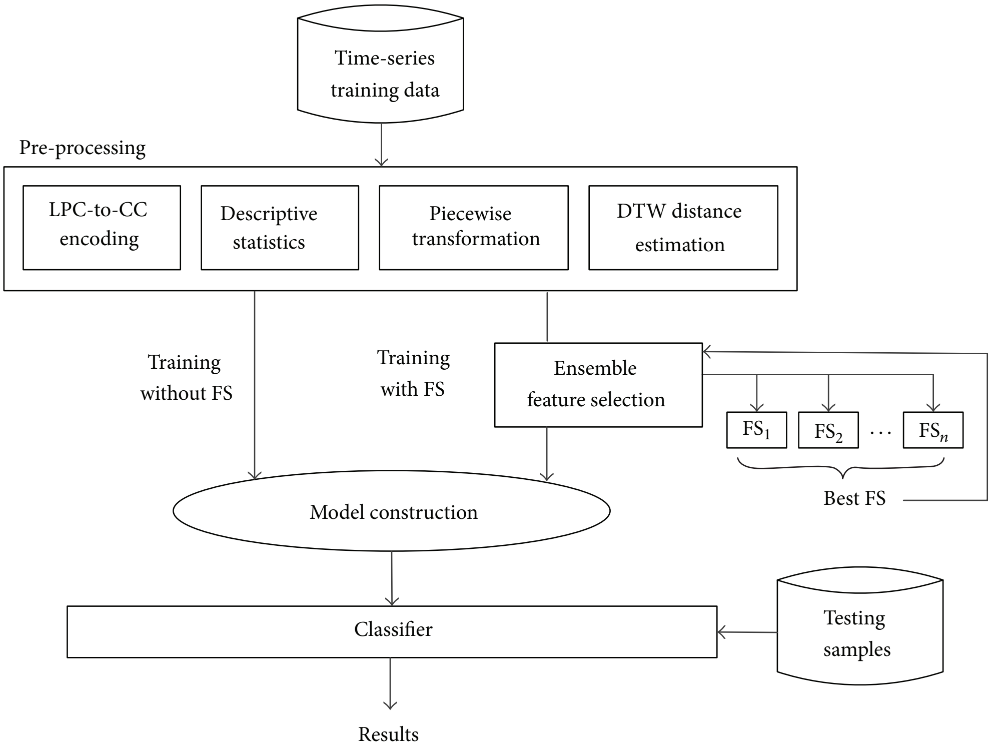


Figure 1: Preprocessing methodology as a part of the classification model learning process.

features. The SFX technique could possibly fit into a standard data mining process, like the one shown in Figure 1. The training dataset in a form of time-series get converted to multidimensional vectors via the preprocessing process, ready to be used for training a classification model. Given a large dimensionality, ensemble feature selection could be applied over the converted multidimensional vectors for refining the accuracy by retaining only some selected relevant features. In our case, a metaheuristic search method seems to perform very well given its efficient stochastic optimization. Its operational nature is dynamic, suitable for choosing features on the fly, considering that voice data could be potentially continuous.

The model construction process is just a standard classification model learning in data mining; for example, a decision tree is built by creating decision paths that map the conditions of the attribute values, as seen from the training samples, to the predicted classes. Once a classifier is trained by processing through the whole training dataset, it is ready to classify new unseen testing samples, and its performance can be measured. The feature selection process is generalized enough to be an ensemble where the winner takes all. During calibration, several feature selection algorithms are put into test, and the best performing one in our case is Feature Selection with Wolf Search Algorithm (FS-WSA) [20]. The other unique contribution by this paper is the extraction of features from the time-series via piece-wise transformation, in addition to the metaheuristic feature selection algorithm. The main difference between our innovation and the others is highlighted in red in Figure 1. We zoom into the details of preprocessing describing the operational flow from data perspective in Figures 2 and 3, respectively, for SFX with and without FS.

In a nutshell, the preprocessing methodology SFX is a way of transforming a two-dimensional time-series (amplitude versus time) into a multidimensional feature vector that has all the essential attributes sufficient to characterize the original time-series voice data. Information is taken from two domains, frequency and time, based on the original time-series. Thus there are two groups of preprocessing techniques being used here, namely, LPC-to-CC encoding (from the frequency domain), Descriptive Statistics of both whole and piecewise, and Dynamic Time Wrap (from the time domain). It is believed that having features obtained from both domains would yield an improved accuracy from the trained classification model due to thorough consideration of the characteristics, hence the representative features, from both domains.

Effectively the preprocessing methodology SFX transforms a matrix of original time-series to a set of training instances which have specific attribute values for building a classification model. Assume V (shown in Figure 2 after the wave read process) is an archive of time-series, with each row containing a jth time-series v_j, and v_j is an ordered sequence of variables $x_j(t)$ such that $v_j = x_j(t) = (x_1, x_2, \ldots, x_m)_j$ where $1 \le t \le m$ is the length of the time-series over different time points and $1 \le j \le n$ is the number of instances in the data archive V.

V is then to be transformed to a structured training dataset S in which each row is an instance s that is defined by a finite number of attributes u, such that $s(j) = (a_1, a_2, \ldots, a_u, Y_j)$ where $1 \le j \le n$ and $1 \le i \le u$. a_i is the ith attribute

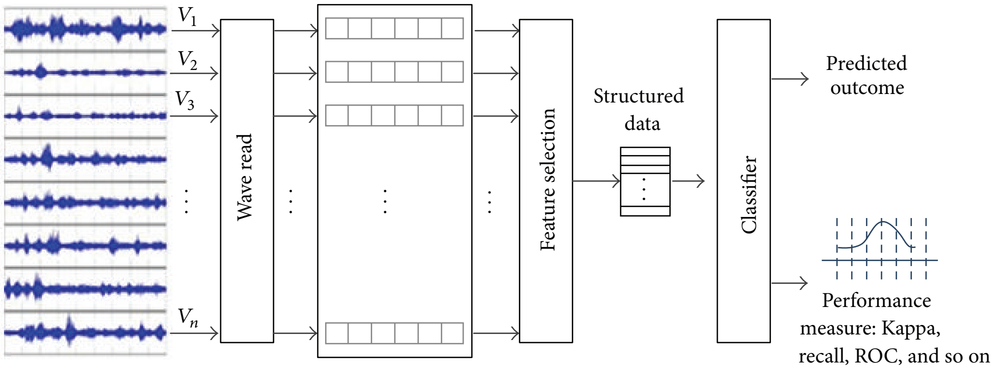


Figure 2: The overall process about SFX.

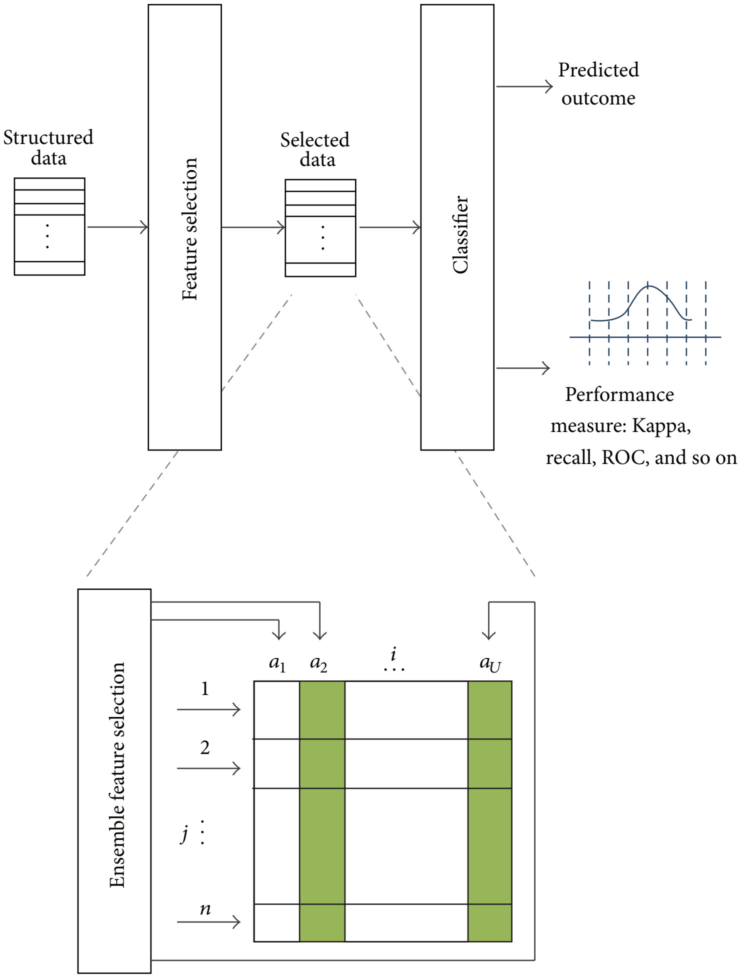


Figure 3: The detailed illustration about SFX with Ensemble FS.

in $s(j)$, Y is a vector of known target values of S; thus Y_j is the jth target value to which the attribute values of $s(j)$ are able to map. The target labels are assumed to be known a-priori in V (supervised learning), and their values are just carried over from V to S, instance for instance, by the same order of j.

The attributes $a_1 \cdots a_u$, however, are obtained from the dual time-frequency domains which can be briefly grouped as $s(j) = [(a_1, a_2, \ldots, a_{uf})_{\text{freq}}, (a_1, a_2, \ldots, a_{ut})_{\text{time}}]$ where the instance $s(j)$ is made of two components that are derived from frequency and time domains, respectively. From the

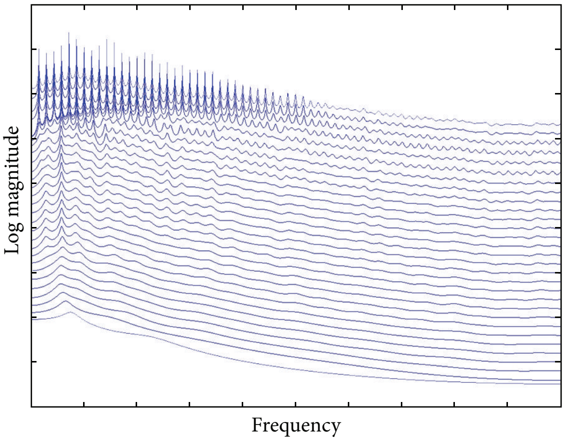


Figure 4: A sample time-series voice data represented in LPC coefficients.

frequency domain alone uf attributes are extracted, ut attributes taken from the time domain, and $u = uf + ut$.

3.1. Feature Extraction from the Frequency Domain. Linear Prediction Coefficients to Cepstral the Coefficients, or Linear Prediction Coding to Cepstrum Coefficients (LPC-to-CC) is selected as the main feature extraction method from the frequency domain in our case. The common production process of human voice contains the following steps of voice generation: the lungs expel air up, acting as the initial step of voice production. Then the air goes into the trachea, passing through the larynx. The larynx is a box-like organ and has two membranes named vocal folds. The voice is actually produced by the vibration of those vocal folds [21]. The acoustic theory of voice production assumes the voice production processes to be a linear system. The output of a linear system is produced based on the linear combination of its previous outputs and current and previous inputs [22]. It is the reason that LPC is chosen here for the purpose of encoding the voice data.

Linear prediction calculates future values of a signal in discrete time format based on a linear function of previous samples. It is always called linear prediction coding, which is a common tool widely used in speech processing for representing the spectral envelope of a signal in compressed form [23].

The original time-series voice data s is windowed by multiplying a windowing sequence $w(n)$ via a hamming method, such that $x(n) = s(n) \otimes w(n)$ where n is the window size. It predicts the next values of points as a linear combination of previous points' values. The predicted points with a pth order of prediction are as follows:

$$\widehat{x}(n) = \sum_{i=1}^{p} a_i \cdot x(n-i), \qquad (1)$$

where a_i is linear predictor coefficients of the ith order. Figure 4 shows a sample of the predictor coefficients.

The problem of value setting of prediction order p determines the characteristics of the vocal filter. If p is too low, then key areas of resonance will disappear; if p is too high, then characteristics of source are missed. Two complex conjugate poles are needed for characterizing correct formants. Thus, in the signal bandwidth, p should be two times of formants number. Suppose f_s is the signal's sampling frequency, and p is usually determined as follows:

$$p = \frac{f_s}{1000} + \gamma, \qquad (2)$$

where γ is the compensation for glottal roll-off and predictor flexibility, which is normally set to be 2 or 3 [24]. The sampling frequency is usually 10 kHz, so the value of p is approximately 12 to 13.

The prediction error generated by this estimate method is the difference between the actual and the predicted values:

$$e(n) = x(n) - \widehat{x}(n) = x(n) - \sum_{i=1}^{p} a_i \cdot x(n-i), \qquad (3)$$

and we define the error metric for the multidimensional signals as

$$e(n) = \left\| x(n) - \widehat{x}(n) \right\| = \sqrt{\sum_{n=-\infty}^{\infty} \left[x(n) - \sum_{i=1}^{p} a_i \cdot x(n-i) \right]^2}. \qquad (4)$$

The expected value of the squared error $E[e^2(n)]$ is minimized, yielding the following equation:

$$R_{ss}(j) = \sum_{i=1}^{p} a_i \cdot R_{ss}(j-i) = \sum_{n=1}^{|S|-1} x(n) \cdot x(n-i), \qquad (5)$$

where $R_{ss}(j)$ is the autocorrelation sequence of signal $x(n)$.

The autocorrelation sequence can then be represented as a matrix in the format of $R \cdot A = -r$ where r is a vector that contains elements of $R(x)$, and A is the vector of predictor coefficients that holds $a(y)$, for $x, y \in [1, p]$. R is known as a Toeplitz Matrix with the size of $p * p$ from which the predictor coefficients can be calculated by inverting the matrix R, $A = -R^{-1}r$. Then the predictor coefficients $A = [a(1), a(2), \ldots, a(p)]$ can be used to derive the cepstrum coefficients, $c(m)$, for $m \in [1, p]$, which are the required output of LPC-to-CC. The cepstrum is defined as the inverse DFT of the log magnitude of the DFT of a signal:

$$c(n) = F^{-1}\left[\log\left|F\{x(n)\}\right|\right], \qquad (6)$$

where F is discrete Fourier transform and F^{-1} is inverted discrete Fourier transform.

When a windowed frame is applied on voice data $y[n]$, the cepstrum is

$$c(n) = \sum_{n=0}^{N-1} \log\left(\left|\sum_{n=0}^{N-1} x(n)\, e^{-j(2\pi/N)kn}\right|\right) e^{j(2\pi/N)kn}. \qquad (7)$$

The transformation steps are shown clearly in Figure 5.

The cepstrum has a lot of advantages such as orthogonality, compactness, and source-filter separation; meanwhile the

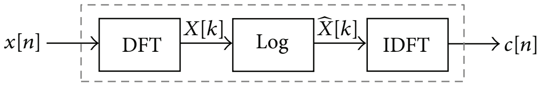


Figure 5: Cepstral Coefficients computation steps.

LPC coefficients are much more susceptible to the precision of numerical numbers, which are less robust than cepstrum coefficients [25]. Thus it is often desirable to transform LPC $\{a_n\}$ into CC $\{c_n\}$:

$$c_n = \begin{cases} \ln G & n = 0, \\ a_n + \dfrac{1}{n}\displaystyle\sum_{k=1}^{n-1} k c_k a_{n-k} & 1 < n \leq p. \end{cases} \tag{8}$$

Above all, the transformation converts the original time-series $x_j(t) = (x_1, x_2, \ldots, x_m)_j$ to a linear prediction coefficient vector defined by $(a_0, a_1, a_2, \ldots, a_{12})_j$ and then converts this vector to a cepstrum coefficient vector defined by $(c_0, c_1, c_2, \ldots, c_{10})_j$. The cepstrum coefficient vector is ready to form a part of the descriptive features, as $(a_1, a_2, a_{uf})_{\text{freq}}$ where $uf = 10$.

3.2. Feature Extraction from the Time Domain. Here we have a feature set $(a_1, a_2, \ldots, a_{ut})_{\text{time}}$ that is characterized by a collection of attribute extracted from the time-series of the voice raw data with respect to the time domain. The statistical attribute extraction method has been commonly used by many researchers in the area of digital signal processing, biosignal analysis, and so forth.

3.2.1. Descriptive Statistics. The extracted statistical features include the following statistics: Mean, Standard Deviation, 1st Quartile, 2nd Quartile, 3rd Quartile, Kurtosis, Interquartile Range, Skewness, RSS (residual sum of squares), Standard Deviation of Residuals, Mean Value of Volatilities, and Standard Deviation of Volatilities. Suppose $X(t)$ is a raw voice data with N sampling points, $R(t)$ is the residual array, and $V(t)$ is the volatility array.

Mean:

$$\overline{X} = \frac{1}{N}\sum_{t=1}^{N} X_t. \tag{9}$$

Standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - \overline{X})^2}. \tag{10}$$

Quartiles: (see Figure 6).

Kurtosis:

$$K = \frac{\sum_{t=1}^{N}\left(X_t - \overline{X}\right)^4}{(N-1)\,\sigma^4}. \tag{11}$$

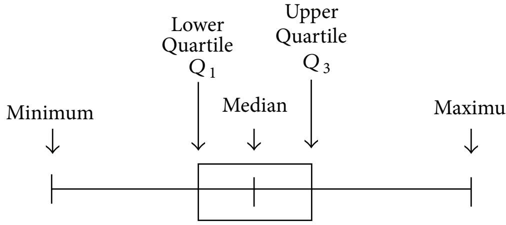


Figure 6: Quartile.

A standard normal distribution has the Kurtosis value of three. As the result, the next definition of kurtosis is widely used and it is often known as excess kurtosis:

$$K = \frac{\sum_{t=1}^{N}(X_t - \overline{X})^4}{(N-1)\,\sigma^4} - 3. \tag{12}$$

Interquartile range:

$$\text{IQR} = \text{Q3} - \text{Q1}. \tag{13}$$

Skewness:

$$S = \frac{\sum_{t=1}^{N}(X_t - \overline{X})^3}{(N-1)\,\sigma^3}. \tag{14}$$

In the statistical analysis of the time-series data, Autoregressive Moving Average models (ARMA) describes a stationary stochastic process based on two polynomials, one for the Auto-regression (AR) and the other for Moving Average (MA) [26]. With the parameter settings this model is usually notated as ARMA(p, q) where p is the order of the AR part and q is the order of the MA part.

Now we introduce another model for characterizing and modeling observed time-series: autoregressive conditional heteroskedasticity (ARCH) model. So that in the model, at any time point in this sequence, it will have a characteristic variance.

If an ARMA model is supposed for the build of error variance, then the model is a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model [27]. With the parameter settings this model is usually referred to as the GARCH(p, q) where p is the order of the GARCH terms σ^2 and q is the order of the ARCH terms ϵ^2:

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + \alpha_1\epsilon_{t-1}^2 + \cdots + \alpha_q\epsilon_{t-q}^2 + \beta_1\sigma_{t-1}^2 + \cdots + \beta_p\sigma_{t-p}^2 \\ &= \alpha_0 + \sum_{i=1}^{q}\alpha_i\epsilon_{t-i}^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i}^2. \end{aligned} \tag{15}$$

We set the parameters of GARCH model with standard values such as the following.

Distribution = "Gaussian";

variance Model = "GARCH";

p (model order of GARCH(p, q)) = "1";

q (model order of GARCH(p, q)) = "1";

r (autoregressive model order of an ARMA(r, m) model) = "1".

RSS:

$$\text{RSS} = \sum_{t=1}^{N} \left(X_t - \widehat{X_t}\right)^2. \quad (16)$$

Standard deviation of residuals:

$$\text{resstd} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left(R_t - \overline{R}\right)^2}. \quad (17)$$

Mean value of volatilities:

$$\text{volmean} = \frac{1}{N} \sum_{t=1}^{N} V_t. \quad (18)$$

Standard deviation of volatilities:

$$\text{volstd} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left(V_t - \overline{V}\right)^2}. \quad (19)$$

3.2.2. Dynamic Time Warping Distance. Though descriptive statistics may give us the overall summary of time-series data and characterize a general shape of time-series data, they may not be able to capture the precise trend movements which are also known as the patterns of evolving lines. In particular we are interested in distinguishing the time-series which belong to one specific class from those that belong to another class. The difference of trend movements can be estimated by a technique called Dynamic Time Warping.

Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two time-series in the situation that both have similar shapes but they vary in time step or speed rate. DTW has been applied to many data objects like video, voice, audio, and graphics. Actually, DTW can explain and deal with any ordered set of data points by the format of linear combination [28].

In theory, DTW is most suitable for voice wave patterns because exact matching for such patterns often may not occur, and voice patterns may vary slightly in the time domain. DTW finds an optimal match between two sequences that allows for compressed sections of the sequences. In other words it allows some flexibility for matching two sequences that may vary slightly in speed or time. The sequences are "warped" nonlinearly in the time dimension to determine a measure of their similarity independent of certain nonlinear variations in the time dimension. Particularly suitable DTW is for matching sequences that may have missing information or various lengths, on condition that the sequences are long enough for matching.

Suppose that $x_j(t)$, $1 \leq j \leq n$ represents an instance in time-series archive X, the number of instances in X is n. c_i, $1 \leq i \leq m$ means each class label to which every instance belongs, where m is the number of class labels. Y_j, $1 \leq j \leq n$ is the jth target value to which the attribute values of $x_j(t)$ are able to map. N_i, $1 \leq i \leq m$ is the number of target values in each class c_i. For any $x(t)$ in time-series archive X, the DTW distance of $x(t)$ to its own class c_i is defined as

$$\text{dist} = \frac{1}{N_i} \sum_{r=1}^{N_i - 1} d_{ir}. \quad (20)$$

Note that the count upper limit is $N_i - 1$ because the DTW distance between $x(t)$ and itself is 0 by the definition (they have the exactly same shape). The DTW distance of $x(t)$ to another class c_j to which it does not belong is

$$\text{dist} = \frac{1}{N_j} \sum_{r=1}^{N_j} d_{ir}. \quad (21)$$

So the number of distance attributes equals the number of c_i, that is, how many classes in total. These distance attributes compose a member of features in $(a_1, a_2, \ldots, a_{ut})_{\text{time}}$, which represents the extracted features of a whole time-series raw data in time domain. Figure 7 visually illustrates this concept of distance in DTW computation.

3.2.3. Piecewise Transformation. So far along the time-domain, statistics are extracted from the whole piece of the time-series as well as the similarity in terms of distance between the test time-series and the mean of its peer group. For a finer level of information, a piecewise transformation is applied which is called Piecewise Linear Function (PLF). A continuous time-series is converted into a collection of linear segments when PLF is applied on it. The purpose of this compressed expression method is to approximate a polynomial curve into a vector of finite n-dimensional Euclidean space that consists of quantitative values.

This is the key part of the research work because it contains our new contribution. Inspired by the financial analysis of stock market, residual and volatility are firstly imported in the application field of voice classification. Like historical volatility for one or more stocks over some specified trading days, we also believe that certain patterns of someone's speech are involved in residual and volatility.

Each sentence is read by *wavread* function in MATHLAB into a one dimension array as illustrated in Figure 2. The starting and ending points of every time-series data are just the same as the beginning and ending points of each array, which means that all information is used without any redundancy. The depth of segmentation n can be selected arbitrarily but sufficiently by the user. In our experiments, the average length of a sentence is ten words, and each word has a peak correspondingly. The mean length of the sampled time-series array is 100 k points. Without compromising the resolution and the complexity of feature space, we choose n to be 20, thus we can cut a peak into two parts which represents up and down gradients. Then the continuous time-series voice data is partitioned equally into 20 pieces.

In our experiment, we try to keep the length of every spoken sentence the same, being almost 10 k points after sampling. The number of segmentations is also 20, so each piece

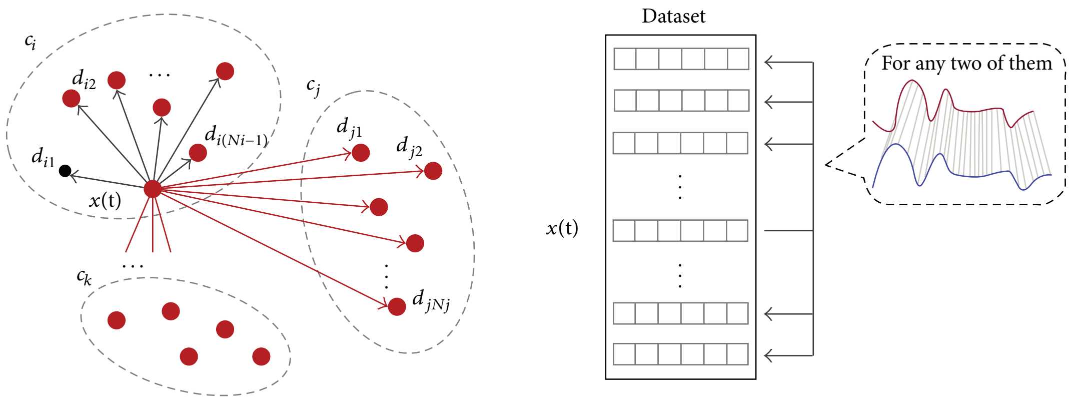


Figure 7: Illustration of DTW calculation.

Table 1: The piecewise segment statistics feature extraction.

Attribute	1	2	3	$\cdots i \cdots$	20
Gradient	Grad 1	Grad 2	Grad 3	······	Grad 20
RSS	RSS 1	RSS 2	RSS 3	······	RSS 20
Resstd	Resstd 1	Rresstd 2	Resstd 3	······	Resstd 20
Volmean	Volmean 1	Volmean 2	Volmean 3	······	Volmean 20
Volstd	Volstd 1	Volstd 2	Volstd 3	······	Volstd 20

maintains at nearly 5 k sampling points. For each segment of the time-series, certain statistics that describe the trend and dynamics of the movement are extracted into the feature vector, that is, $(a_1, a_2, \ldots, a_{ut})_{\text{time}}$. An example of the time-series segmentation in normal and stretched view is shown in Figure 8.

Using this piecewise method, the features that are being extracted are statistics of each partition of the time-series. Table 1 shows a list of all statistics that can potentially be harvested from 20 partitions of a particular time-series. The definitions of the statistics parameters then follow.

For each segment $s_i(t)$, $1 \leq i \leq 20$, n is the number of points each segment contains, that is, $n = |s_i(t)|$.

Gradient of $s_i(t)$:

$$\text{grad}_i = \beta_i, \tag{22}$$

where β_i is

$$s_i = \beta_i t + \alpha_i + \varepsilon_i. \tag{23}$$

RSS of $s_i(t)$:

$$\text{RSS}_i = \sum_{t=1}^{n} \left(s_{it} - \widehat{s_{it}}\right)^2. \tag{24}$$

Standard deviation of residuals of $s_i(t)$:

$$\text{resstd}_i = \sqrt{\frac{1}{n}\sum_{t=1}^{n} \left(\varepsilon_{it} - \overline{\varepsilon}_i\right)^2}. \tag{25}$$

Mean value of volatilities of $s_i(t)$:

$$\text{volmean}_i = \frac{1}{n}\sum_{t=1}^{n} V_{it}. \tag{26}$$

Standard deviation of volatilities of $s_i(t)$:

$$\text{volstd}_i = \sqrt{\frac{1}{n}\sum_{t=1}^{n} \left(V_{it} - \overline{V_i}\right)^2}. \tag{27}$$

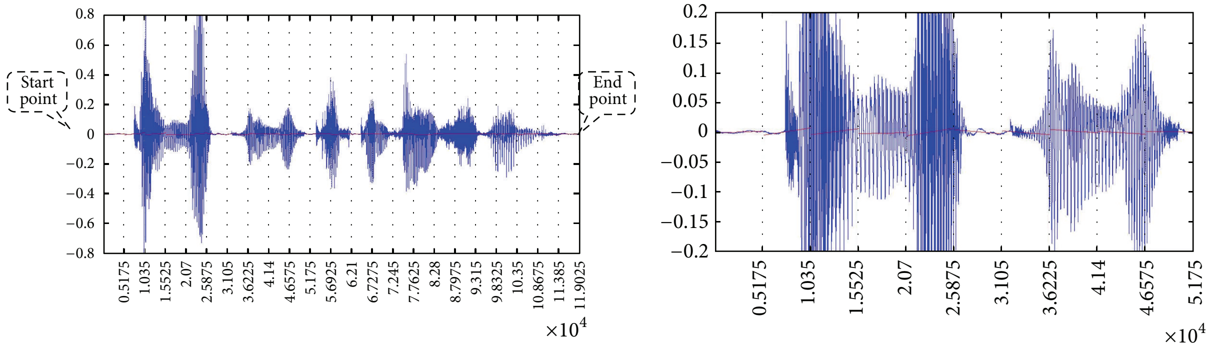
The model for residual and volatility is also selected as GARCH model, where the parameters of GARCH model are configured the same as previously: mentioned Distribution = "Gaussian"; Variance Model = "GARCH"; p (model order of GARCH(p, q)) = "1"; q (model order of GARCH(p, q)) = "1"; r (autoregressive model order of an ARMA(r, m) model) = "1".

A calibration test is used to determine the optimal choice of the length of each piece (interval) such that the highest classification accuracy can be obtained. Different numbers of intervals have been tried continually for piecewise transformation, extracting the corresponding attributes and running the classifiers. As the results shown in Figure 9, it was found that using 20 segments of each length yields the highest classification accuracy. The test was done preliminarily without FS and the results are averaged over all parameters.

4. Experiment

In order to compare the effectiveness of the proposed time-series preprocessing method with the other existing methods, we test them on four different voice/speech datasets using nearly twenty popular and traditional classification algorithms in data mining.

4.1. Data Description. Four representative types of voice data are tested by the simulation experiments; they are Female and Male (FM) Dataset, Emotional Speech (ES) Dataset, Speaker Identification (SI) Dataset, and Language Recognition (LR) Dataset.



(a)

(b)

FIGURE 8: (a) An example of sampled time-series voice data and its partition. (b) The amplified view of piecewise linear regression (partly).

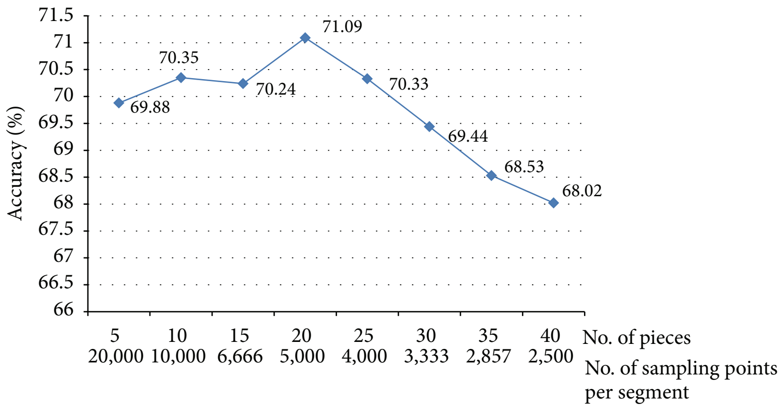


FIGURE 9: Calibration curve for segmentation selection.

4.1.1. Data Sources

FM. The FM dataset is downloaded from School of Information Technology and Electrical Engineering (ITEE), University of Queensland, Australia, called VidTIMIT Audio-Video Dataset [29]. The dataset is made up of audio recordings of recited short sentences from 43 volunteers, among which 19 are females and 24 are males. It is from the test section of TIMIT corpus that all those sentences were selected. 10 sentences for every speaker. The first two sentences are all the same for each speaker, with the remaining eight that differ according to every individual. Here only the audio data is concerned and video data is discarded.

ES. The ES dataset comes from the database of German emotional speech, developed at the Technical University, Institute for Speech and Communication, Department of Communication Science, Berlin, with Professor Sendlmeier. It was funded by the German Research Association DFG (research project SE 462/3-1) [30]. The aim of the database is to examine acoustical correlates of emotional speech. It is comprised of seven basic emotions (anger, happiness, sadness, fear, disgust, boredom, and neutral) and only four major emotions are taken for the purpose of simplification. Ten professional native German actors with balance gender distribution (5 for each) produced these emotional speeches, which containing 10 sentences with 5 short sentences and 5 longer ones.

SI. The SI dataset is taken from the PDA speech database, owned by Yasunari Obuchi in March 2003, Carnegie Mellon University (CMU). The recording was done by CMU students and staff [31]. There recording was done by using one PDA with four small microphones mounted around and one big microphone in the record room. The type of that big microphone was an Optimus Nova 80 close-talk microphone. The type of small ones was Panasonic WM-55DC2 and they were mounted using a mock-up shown below. There are 16 speakers and each read about 50 sentences.

LR. The LR dataset is generated through an approach called speech synthesis. The speech synthesizer software used here is Microsoft Text-to-Speech engine with many expansion packages [32]. Sentences of English, Cantonese, and Mandarin were widely selected from the area of frequently used daily conversations, daily news, educational reports, stories, scientific articles, ancient proses, poems and poetries, and so forth.

4.1.2. Data Formats. The voice data is in the format of two-dimensional time-series, with an amplitude value in sound that varies over time; examples are given in Figures 8(a) and 8(b). The sampling rate or frequency of wave read process is 10 kHz. Group distributions of distinctive datasets are given in Table 2. The FM dataset has only two classes, which is the simplest classification task in data mining. The rest of datasets

Table 2: Distributions of classes in different datasets.

Dataset name	No. of classes or labels	Notes
FM	2	Female and male
ES	4	Happiness, anger, sadness, and neutral
SI	16	16 different speakers
LR	3	Cantonese, English, and Mandarin

Table 3: The numbers of attributes associated with datasets and instances for training and testing by various preprocessing methods.

Preprocessing method	FM	ES	SI	LR
Wavelet	50	50	50	50
LPC-to-CC	10	10	10	10
SFX	74	68	88	75
SFX + FS	20	53	20	32
No. of instances for training	258	179	564	600
No. of instances for testing	172	160	272	150

contain more than two classes that make the classification task more difficult. The numbers of attributes or features for every dataset and instances for training and testing are listed in Table 3.

4.1.3. Data Visualization. Visualization of parts of each group of the datasets, FM, ES, SI, and LR is displayed in Figures 10(a) to 10(l). Inspecting by just naked eyes, one can see some distinctive differences between the waveforms of different classes.

Multidimensional (MD) visualization of each group of those datasets is shown in Figures 11(a) to 11(b). Again, by just visual inspection, it can be observed that the voice data between different classes are apparently distinctive in the FM group and in the LR group. Common sense tells us that female speakers and male speakers have distinguishing vocal tones. Speeches of different languages also can be differentiated easily, as each language has its unique vowels and phonics. In contrast, the voice data of 16 unique speakers have certain overlaps in their feature values; this implies that some speakers share similar voices which are not something very uncommon in real life. The voice data in the emotion groups are highly mixed together by the feature values. That shows the potential computational difficulty in classification between voices of different emotions.

4.1.4. Algorithms Used in Comparison. Our experiments are performed by using popular and standard classification algorithms (with their default parameters applied) over the four sets of the above-mentioned voice data that are being handled by four preprocessing methods. A total of 20 classification algorithms are being used. The justification is that we try to test the generality of our voice classification model without being attached to any specific classification algorithm. In other words, the design of the voice classification model should be generic enough, and its efficacy should be independent from the choice of classifier. While the focus of the voice classification model is centered at the preprocessing steps which leverage the features from both time and frequency domains followed by feature selection for reducing the feature space dimension, classification algorithms can become flexible plug-and-play in our model design. The standard classification algorithms used in our experiments are well known in data mining research community as well as available in Weka (http://www.cs.waikato.ac.nz/ml/weka/), and they are listed in Table 4.

Table 4: List of standard classification algorithms used in our experiment.

Standard classification algorithm type	Algorithm
Bayes	NaiveBayes
Functions	LibSVM Multilayer perceptron SMO
Meta	Bagging
Rules	Conjunctive rule Decision table FURIA JRip/RIPPER NNge OneR PART
Decision Trees	BF tree FT J48/C4.5 LMT NB tree Random forest Random tree REP tree

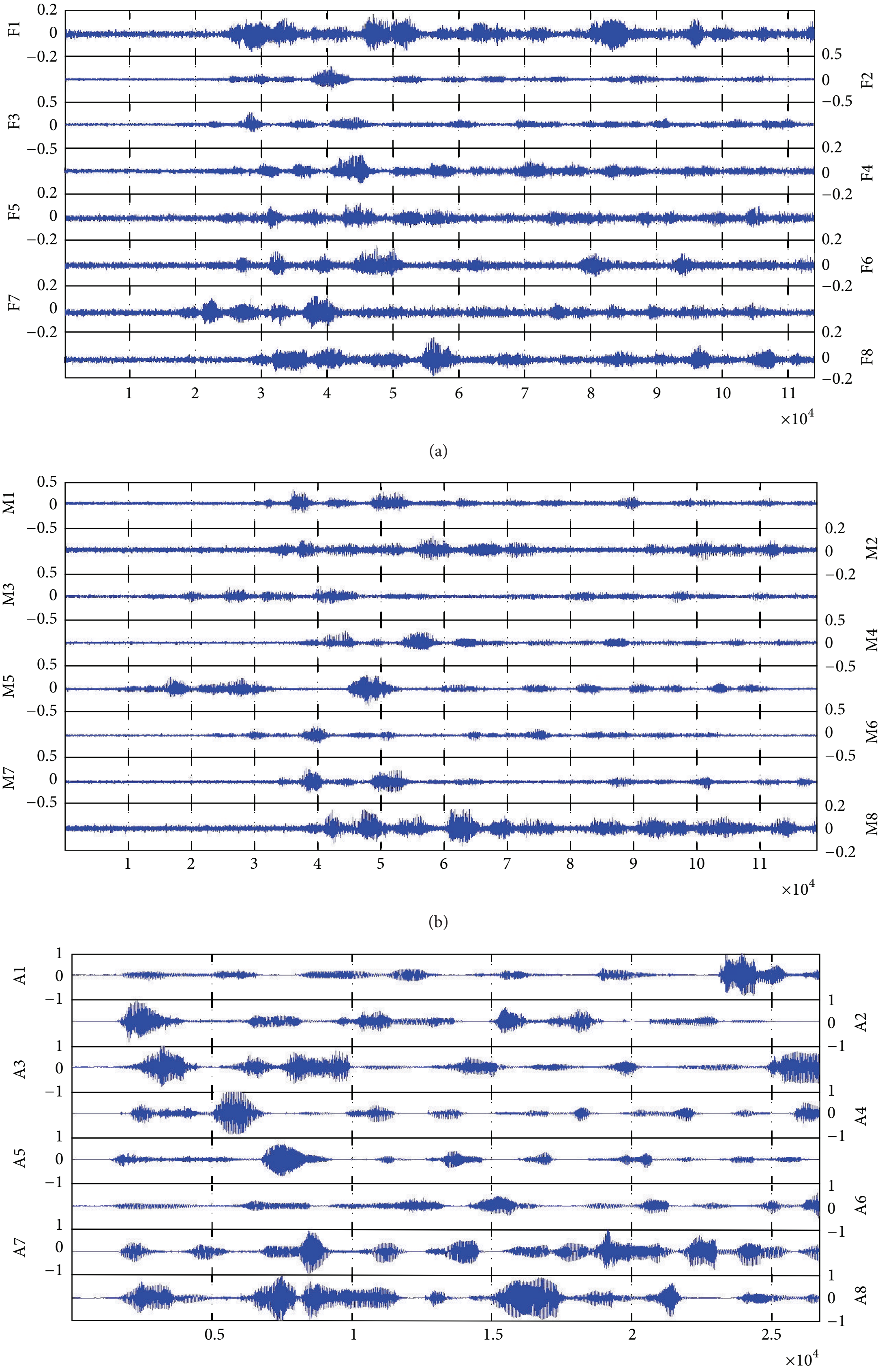
The four preprocessing methods used for comparison are as follows.

LPC-to-CC. Only the cepstrum coefficients are used as the encoding result of time-series voice data. Meanwhile, the LPC coefficients are ignored in final attributes set.

Wavelet. Only the 50-largest Harr wavelet coefficients are taken as converting the sequence from time domain to frequency domain. The number of decomposition level of Harr wavelet transform is 3.

SFX. Statistical Feature Extraction (SFX) converts the time-series voice data to a whole set of attributes with both frequency and time domains, using a collection of feature methods described in Section 3.

SFX + FS. Statistical Feature Extraction + Feature Selection (SFX + FS) is exactly the same as SFX except that the full set of features or attributes were filtered by using different

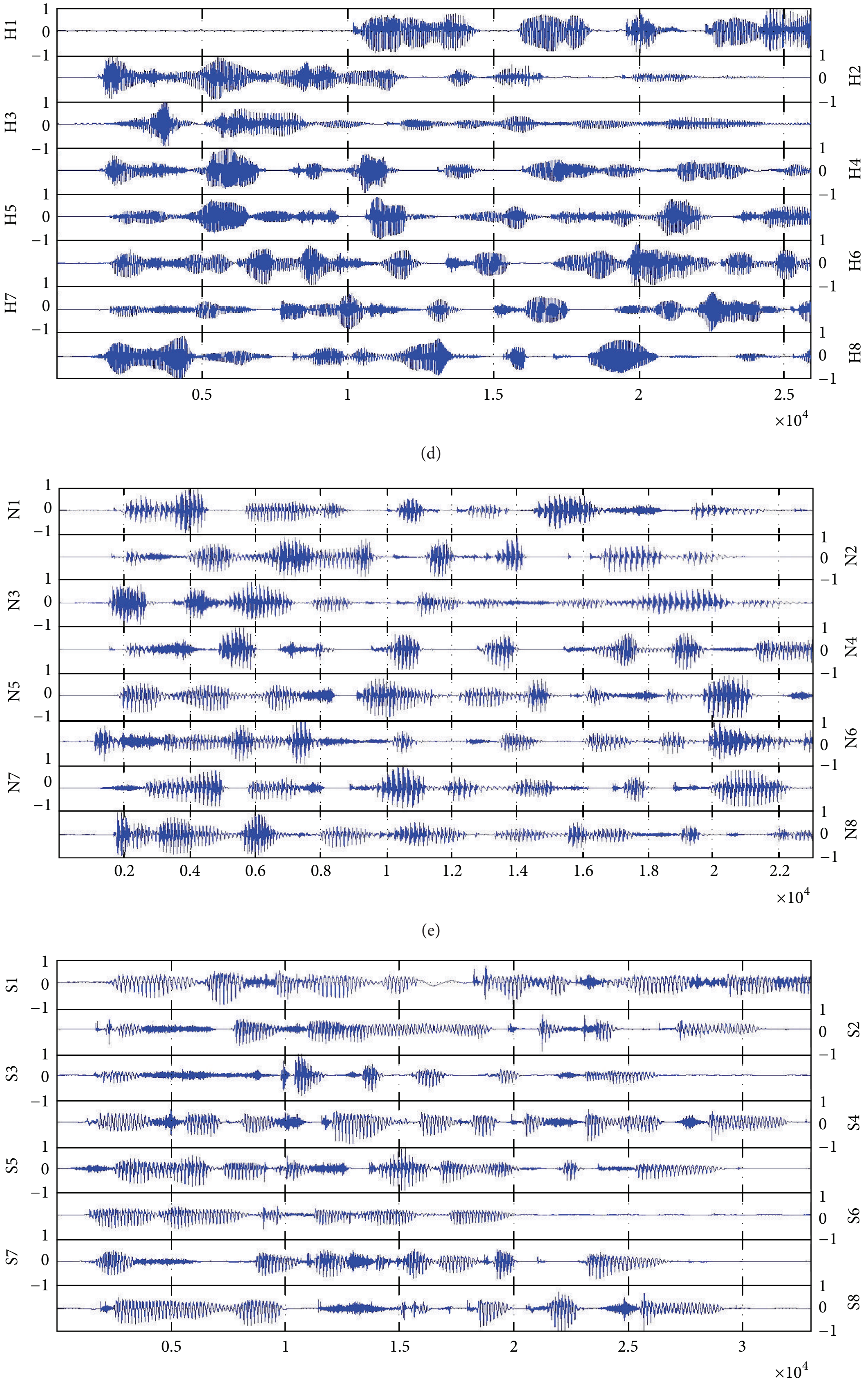


(a)

(b)

(c)

Figure 10: Continued.

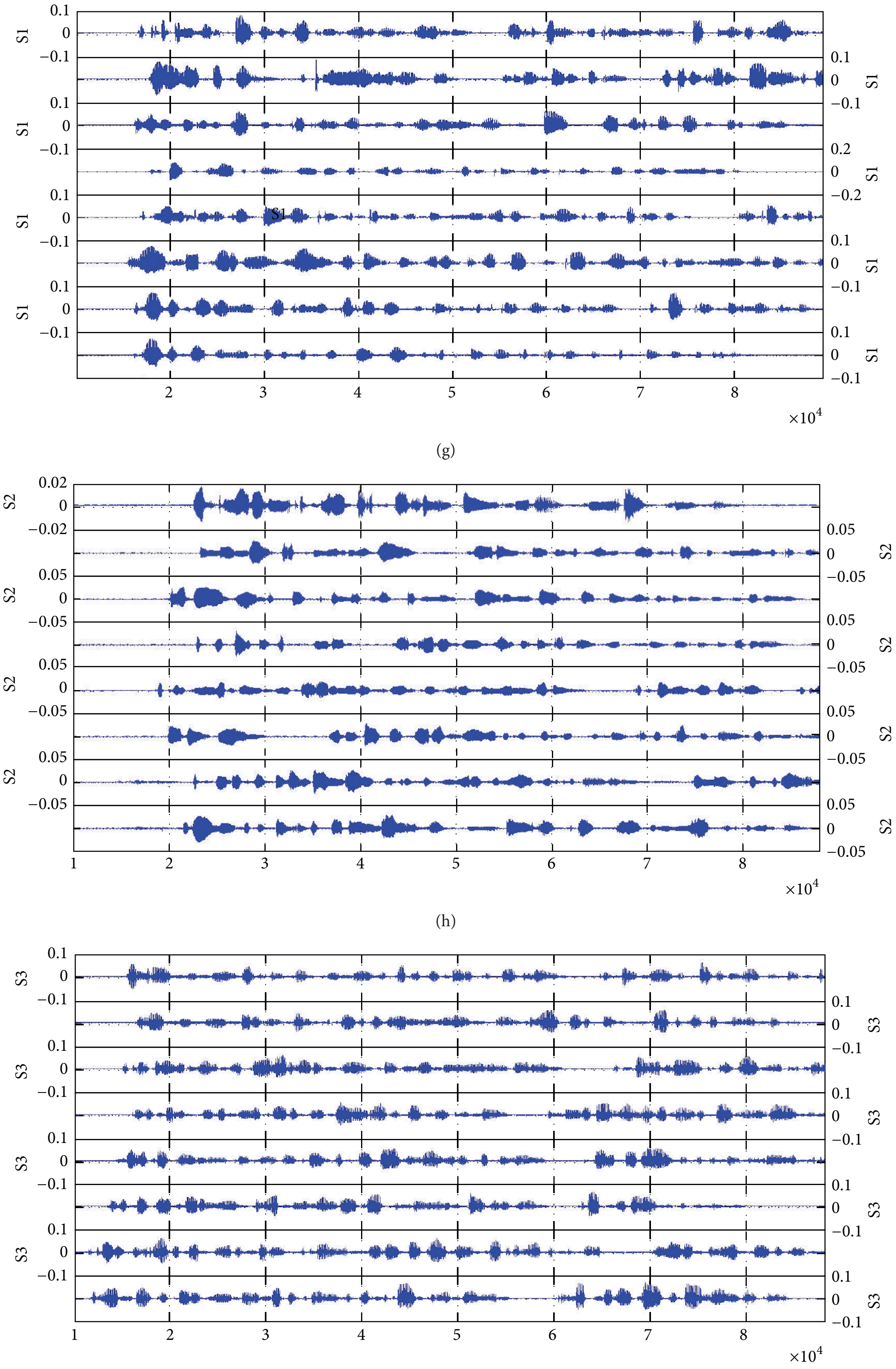


(d)

(e)

(f)

Figure 10: Continued.

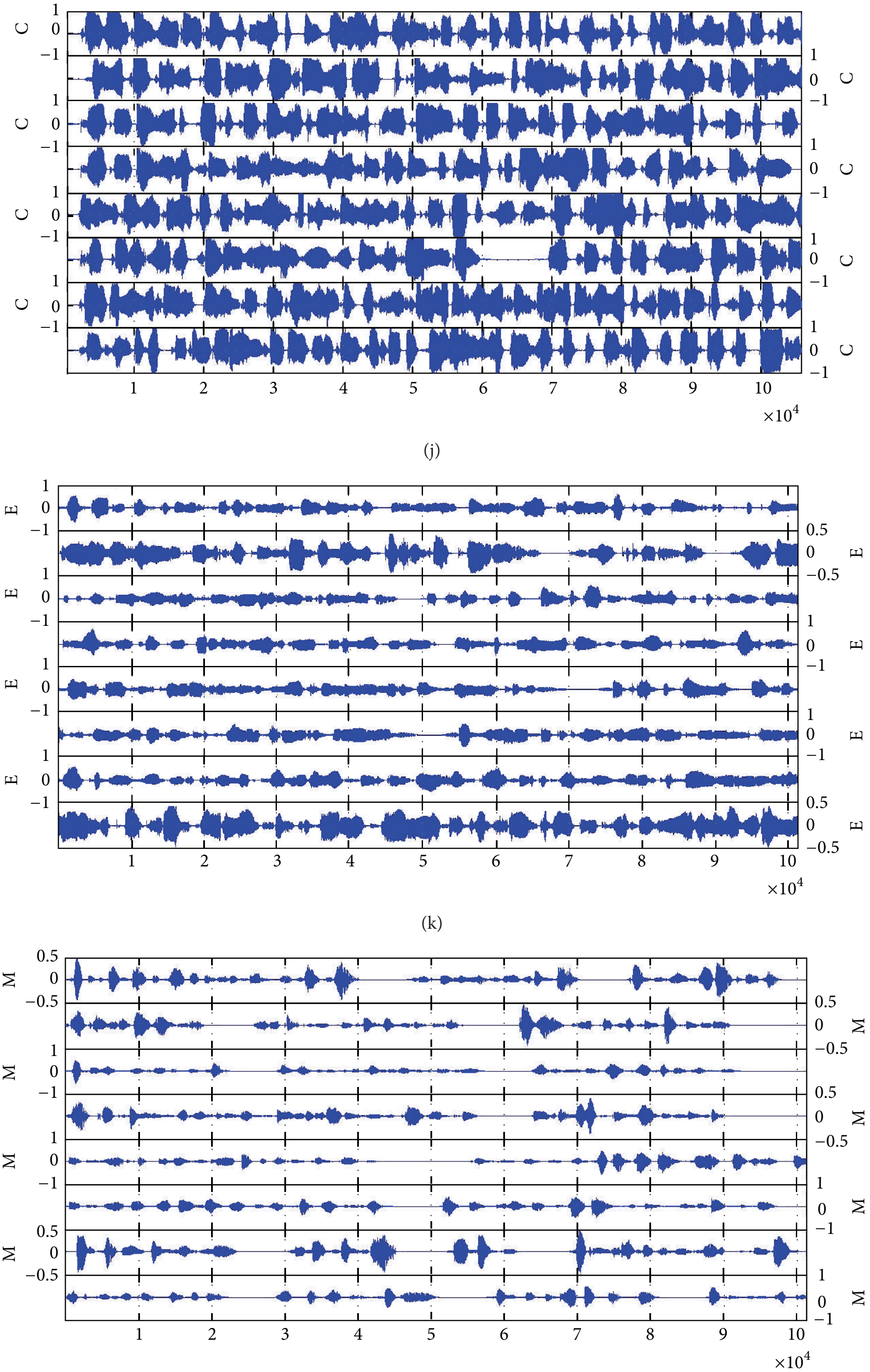


(g)

(h)

(i)

Figure 10: Continued.



(j)

(k)

(l)

Figure 10: (a) Visualization of FM dataset that belongs to the "Female" group. (b) Visualization of FM dataset that belongs to the "Male" group. (c) Visualization of ES dataset that belongs to the "Anger" group. (d) Visualization of ES dataset that belongs to the "Happiness" group. (e) Visualization of ES dataset that belongs to the "Neutral" group. (f) Visualization of ES dataset that belongs to the "Sadness" group. (g) Visualization of SI dataset that belongs to the "Speaker 1" group. (h) Visualization of SI dataset that belongs to the "Speaker 2" group. (i) Visualization of SI dataset that belongs to the "Speaker 3" group. (j) Visualization of LR dataset that belongs to the "Cantonese" group. (k) Visualization of LR dataset that belongs to the "English" group. (l) Visualization of LR dataset that belongs to the "Mandarin" group.

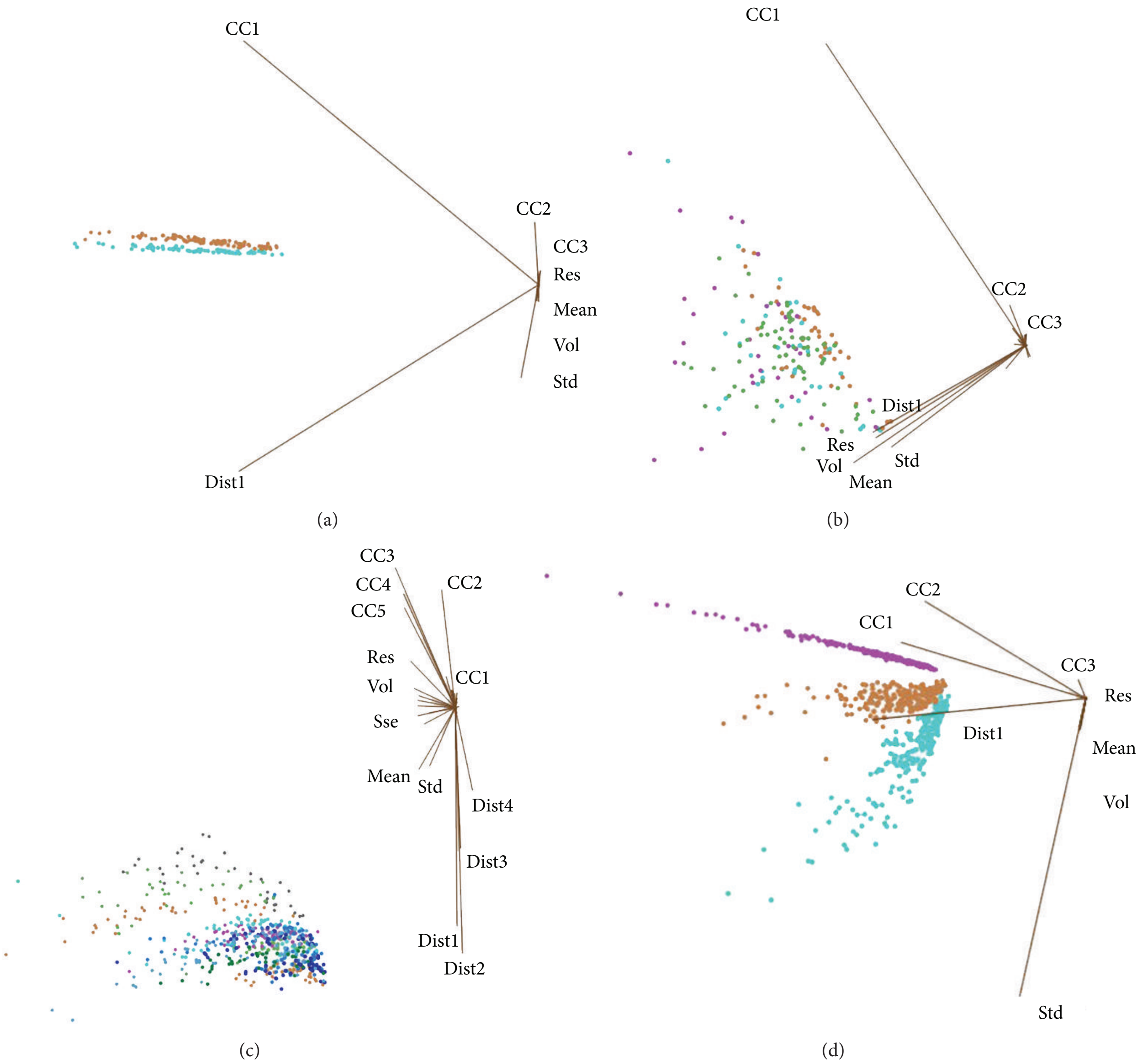


Figure 11: (a) MD visualization of FM. (b) MD visualization of ES. (c) MD visualization of SI. (d) MD visualization of LR.

feature reduction methods. Note that it is an ensemble feature selection method, using multiple models to obtain the best performance. Two facts are considered: mean accuracy and time cost. The compensation is made between time and accuracy, which means that we prefer a little bit lower accuracy and more on acceptable time cost. The optimal one was chosen as the final FS method.

WSA. Wolf Search Algorithm (WSA) is a bioinspired heuristic optimization algorithm [33]. It naturally balances scouting the problem space in random groups (breadth) and searching for the solution individually (depth). The pseudocode of WSA is given in Pseudocode 1.

Chi-Square. In statistics, the purpose of chi-square (χ^2) test is to measure the independence of two events A and B. From the knowledge of probability and statistics, we know that two events are independent if the probability equation has the following relationships: $P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$ or $P(AB) = P(A)P(B)$ equivalently. In feature selection, let occurrence of the term be event A and occurrence of the class be event B. We then rank values based on the following quantity [34, 35]:

$$\chi^2(D,t,c) = \sum_{e_t\in\{0,1\}} \sum_{e_c\in\{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}, \qquad (28)$$

where D is the whole set of observations, N is the frequency actually found in D, and E is the expected one. At the same time, $e_t = 1$ means that the document contains term t, $e_t = 0$ means that the document does not contain t, $e_c = 1$ means

```
Objective function $f(x)$, $x = (x_1, x_2, \ldots, x_d)^T$
Initialize the population of wolves, $x_i$ for $i = 1, 2, \ldots, W$
Define and initialize parameters:
$r$ = radius of the visual range
$s$ = step size by which a wolf moves at a time
$\alpha$ = velocity factor of wolf
$P_e$ = a user-defined threshold [0, 1], that determines how frequently an enemy appears, so will the wolf escape
WHILE ($t$ < generations && stopping criteria not met)
  FOR $i = 1, \ldots, W$   //   for each wolf
      Prey_new_food_initiatively ();
      Generate_new_location ();
      // check whether the next location suggested by the random number generator is new. If not, repeat
       generating random location.
      IF(dist($x_i, x_j$) < $r$ && $x_i$ is better as $f(x_1) < f(x_j)$)
        $x_i$ moves towards $x_j$   //   $x_j$ is at a better place than $x_i$
      ELSE-IF
        $x_i$ = Prey_new_food_passively ();
      END-IF
      Generate_new_location ();
      IF (rand () > $p_e$)
        $x_i = x_i$ + rand () + $v$;   //   escape to a new position farther than $v$
      END-IF
  END-FOR
END-WHILE
```

Pseudocode 1: Pseudocode of WSA.

that the document is in class c, and $e_c = 0$ means that the document is not in c.

CFS. An essential assumption is made before going directly into the discussion of Correlation Feature Selection (CFS). It is that good feature subsets always have highly corresponding features, whereas there are uncorrelated features among the rest of them [36]. On the basis of that, CFS starts its work and evaluates features. The merit containing k features for a specific feature subset S is

$$\text{Merit}_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \qquad (29)$$

where $\overline{r_{cf}}$ represents the average value of all c-f (classification to feature) correlations, and $\overline{r_{ff}}$ is the mean value of all f-f (feature to feature) correlations. Then CFS is defined as follows:

$$\text{CFS} = \max_{S_k} \frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_k}}{\sqrt{k + 2\left(r_{f_1 f_2} + \cdots + r_{f_i f_j} + \cdots + r_{f_k f_1}\right)}}, \qquad (30)$$

where r_{cf_i} and $r_{f_i f_j}$ variables are correlations just like the aforementioned.

MRMR. Maximum Relevance is normally referred to as subsets of data identified by feature selection which are relevant to the parameters. There often exist relevant but redundant components in those subsets. MRMR, known as Minimum Redundancy Maximum Relevance, however, attempts to detect those redundant subsets, find them out, and delete them. Example application fields of MRMR are but not limited to cancer diagnosis, face detection, autoresponse, and speech recognition.

Suppose $p(x)$, $p(y)$, and $p(x, y)$ to be probabilistic density functions of two random variables x and y; then their mutual information is defined as [37]

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)\, p(y)} dx\, dy. \qquad (31)$$

The nature of feature selection in mutual information model is to find a feature set S containing m features $\{x_i\}$, which also have the largest dependency on the target class c. This is the definition of Max Dependency:

$$\max D(S, c), \quad D = I(\{x_i, i = 1, 2, \ldots, m\}; c). \qquad (32)$$

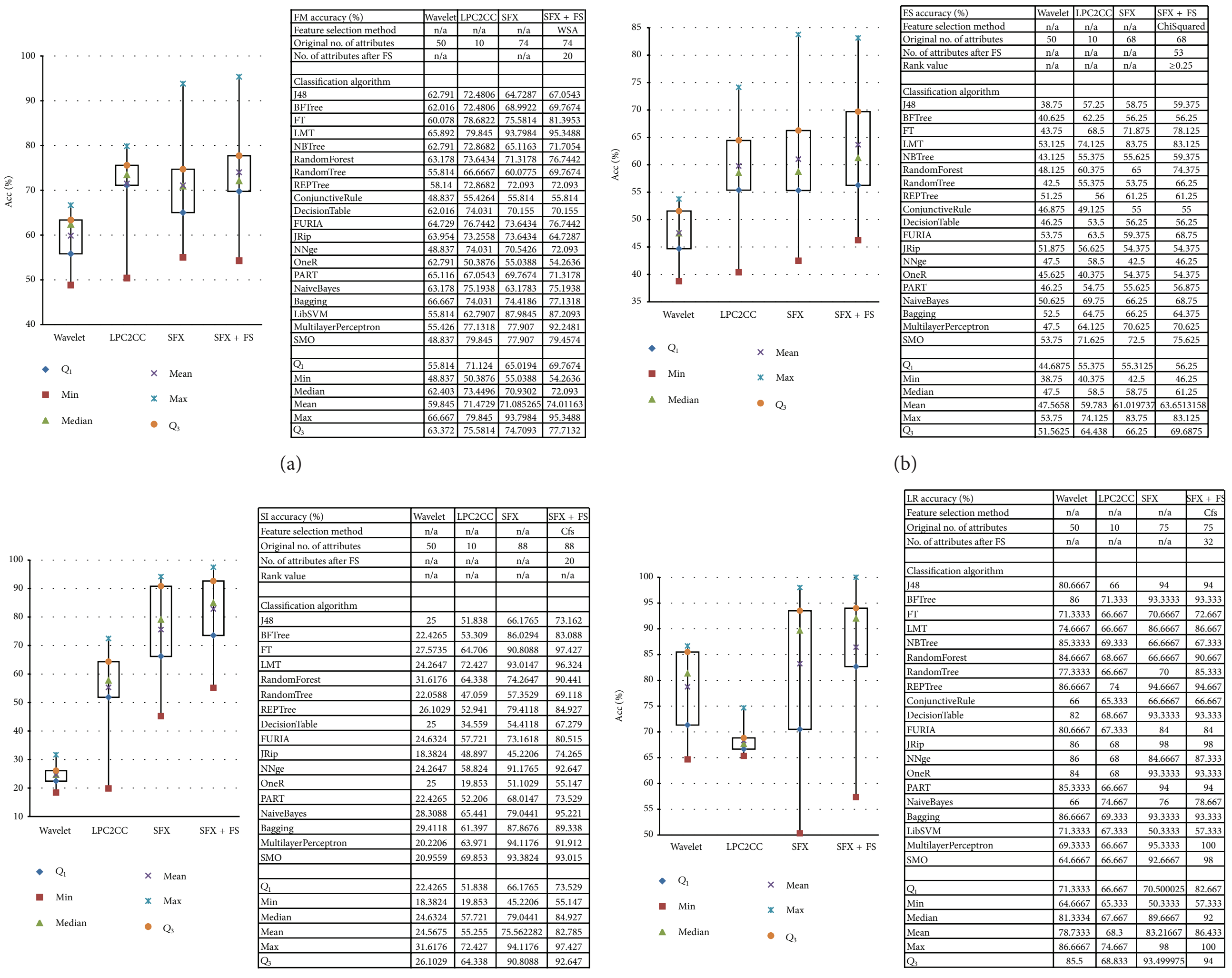
Max-Relevance and Min-Redundancy are

$$\begin{aligned} &\max D(S, c), \quad D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c), \\ &\min R(S), \quad R = \frac{1}{|S|^2} \sum_{x_i, x_i \in S} I(x_i, x_j). \end{aligned} \qquad (33)$$

Out of the chosen popular feature selection algorithms that are put into test in the calibration process, we can see that WSA which is a metaheuristic FS algorithm consistently is having superior performance, except for the Speaker Identification dataset which is known for its overlaps in feature values. The testing results are shown in full in Table 5. The computing environment is on a PC workstation, with Windows 7 Enterprise Edition, 64 bits, Intel Core i7 CPU, and 8 GB RAM.

Table 5: Optimal FS methods for each dataset.

FS accuracy %	FM				ES				SI				LR			
Feature selection method	CFS	ChiS	MRMR	WSA	CFS	ChiS	MRMR	WSA	CFS	ChiS	MRMR	WSA	CFS	ChiS	MRMR	WSA
Original no. of attributes	74	74	74	74	68	68	68	68	88	88	88	88	75	75	75	75
No. of attributes after FS	8	55	30	20	17	53	30	21	20	71	30	29	32	32	30	31
Classification algorithm																
J48	63.1783	64.7287	63.5659	67.4419	61.25	59.375	57.5	64.375	73.1618	66.1765	74.6324	64.7059	94	92.3333	70	94
BFTree	71.7054	68.9922	62.0155	76.3566	58.125	56.25	65	65.625	83.0882	79.7794	85.6618	81.25	93.3333	91.6666	68	93.3333
FT	74.031	79.0698	72.8682	82.5581	64.375	78.125	65.625	88.75	97.4265	88.9706	93.75	88.2353	72.6667	71	69.3333	76
LMT	63.5659	92.4419	73.6434	97.6744	60.625	83.125	70	88.125	96.3235	92.4465	91.9118	87.8676	86.6667	85	66.6667	86.6667
NBTree	71.7054	63.1783	63.1783	70.155	51.25	59.375	67.5	66.25	n/a	n/a	n/a	n/a	67.3333	65.6666	66.6667	68
RandomForest	73.6434	70.5426	67.8295	72.8682	61.875	74.375	71.25	73.125	90.4412	74.2647	81.9853	81.9853	90.6667	89	70.6667	90.6667
RandomTree	64.7287	59.6899	55.814	70.155	45	66.25	57.5	67.5	69.1176	61.7647	69.4853	63.9706	85.3333	83.6666	84.6667	85.3333
REPTree	72.093	72.093	67.4419	73.6434	56.875	61.25	60	64.375	84.9265	84.1912	79.4118	83.4559	94.6667	93	66.6667	94.6667
ConjunctiveRule	55.814	55.814	48.8372	65.8915	54.375	55	55	56.25	n/a	n/a	n/a	n/a	66.6667	65	65.3333	66.6667
DecisionTable	70.155	70.155	53.876	68.6047	57.5	56.25	61.875	52.5	67.2794	58.8235	63.2353	59.5588	93.3333	91.6666	77.3333	93.3333
FURIA	71.7054	77.1318	63.9535	74.8062	62.5	68.75	64.375	53.125	80.5147	62.5	78.6765	86.3971	84	82.3333	70	84
JRip	72.8682	71.7054	67.8295	73.6434	66.25	54.375	58.125	55	74.2647	34.9265	72.0588	65.4412	98	96.3333	66.6667	98
NNge	68.9922	66.6667	55.814	63.5659	53.125	46.25	52.5	50.625	92.6471	91.1765	92.2794	81.25	87.3333	85.6666	69.3333	87.3333
OneR	50.3876	50.3876	54.2636	54.2636	54.375	54.375	54.375	61.25	55.1471	55.1471	49.2647	49.2647	93.3333	91.6666	54	93.3333
PART	64.3411	68.6047	62.4031	68.6047	58.75	56.875	58.125	70.625	73.5294	67.2794	80.5147	71.3235	94	92.3333	69.3333	94
NaiveBayes	70.9302	64.7287	67.4419	67.0543	64.375	68.75	58.75	58.75	95.2206	76.8382	86.0294	72.0588	78.6667	77	64	78.6667
Bagging	73.2558	75.1938	70.155	75.969	63.75	64.375	68.75	53.125	89.3382	85.6618	84.9265	86.3971	93.3333	91.6666	66.6667	94.6667
LibSVM	66.6667	87.5969	63.5659	89.9225	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	57.3333	55.6666	46.6667	57.3333
MultilayerPerceptron	68.6047	79.0698	72.4806	89.5349	63.75	70.625	61.875	66.875	91.9118	93.75	93.75	86.3971	100	98.3333	70	100
SMO	72.4806	77.907	73.6434	79.4574	76.875	75.625	61.875	67.5	93.0147	93.75	93.3824	80.5147	98	96.3333	76	99.3333
Mean accuracy %	68.04263	70.78489	64.03102	**74.108535**	59.73684	63.65132	61.57895	**64.40789**	82.78547	74.55568	80.64448	*75.88668*	86.43333	84.76663	67.90001	**86.76667**
Time (s)	0.78	2.867	3.56	*31.275*	1.03	3.328	1.439	***441.476***	1.91	3.815	3.26	***3585***	1.39	2.17	4.8	***906***



FM accuracy (%)	Wavelet	LPC2CC	SFX	SFX + FS
Feature selection method	n/a	n/a	n/a	WSA
Original no. of attributes	50	10	74	74
No. of attributes after FS	n/a		n/a	20
Classification algorithm				
J48	62.791	72.4806	64.7287	67.0543
BFTree	62.016	72.4806	68.9922	69.7674
FT	60.078	78.6822	75.5814	81.3953
LMT	65.892	79.845	93.7984	95.3488
NBTree	62.791	72.8682	65.1163	71.7054
RandomForest	63.178	73.6434	71.3178	76.7442
RandomTree	55.814	66.6667	60.0775	69.7674
REPTree	58.14	72.8682	72.093	72.093
ConjunctiveRule	48.837	55.4264	55.814	55.814
DecisionTable	62.016	74.031	70.155	70.155
FURIA	64.729	76.7442	73.6434	76.7442
JRip	63.954	73.2558	73.6434	64.7287
NNge	48.837	74.031	70.5426	72.093
OneR	62.791	50.3876	55.0388	54.2636
PART	65.116	67.0543	69.7674	71.3178
NaiveBayes	63.178	75.1938	63.1783	75.1938
Bagging	66.667	74.031	74.4186	77.1318
LibSVM	55.814	62.7907	87.9845	87.2093
MultilayerPerceptron	55.426	77.1318	77.907	92.2481
SMO	48.837	79.845	77.907	79.4574
Q_1	55.814	71.124	65.0194	69.7674
Min	48.837	50.3876	55.0388	54.2636
Median	62.403	73.4496	70.9302	72.093
Mean	59.845	71.4729	71.085265	74.01163
Max	66.667	79.845	93.7984	95.3488
Q_3	63.372	75.5814	74.7093	77.7132

(a)

ES accuracy (%)	Wavelet	LPC2CC	SFX	SFX + FS
Feature selection method	n/a	n/a	n/a	ChiSquared
Original no. of attributes	50	10	68	68
No. of attributes after FS	n/a	n/a	n/a	53
Rank value	n/a	n/a	n/a	≥0.25
Classification algorithm				
J48	38.75	57.25	58.75	59.375
BFTree	40.625	62.25	56.25	56.25
FT	43.75	68.5	71.875	78.125
LMT	53.125	74.125	83.75	83.125
NBTree	43.125	55.375	55.625	59.375
RandomForest	48.125	60.375	65	74.375
RandomTree	42.5	55.375	53.75	66.25
REPTree	51.25	56	61.25	61.25
ConjunctiveRule	46.875	49.125	55	55
DecisionTable	46.25	53.5	56.25	56.25
FURIA	53.75	63.5	59.375	68.75
JRip	51.875	56.625	54.375	54.375
NNge	47.5	58.5	42.5	46.25
OneR	45.625	40.375	54.375	54.375
PART	46.25	54.75	55.625	56.875
NaiveBayes	50.625	69.75	66.25	68.75
Bagging	52.5	64.75	66.25	64.375
MultilayerPerceptron	47.5	64.125	70.625	70.625
SMO	53.75	71.625	72.5	75.625
Q_1	44.6875	55.375	55.3125	56.25
Min	38.75	40.375	42.5	46.25
Median	47.5	58.5	58.75	61.25
Mean	47.5658	59.783	61.019737	63.6513158
Max	53.75	74.125	83.75	83.125
Q_3	51.5625	64.438	66.25	69.6875

(b)

SI accuracy (%)	Wavelet	LPC2CC	SFX	SFX + FS
Feature selection method	n/a	n/a	n/a	Cfs
Original no. of attributes	50	10	88	88
No. of attributes after FS	n/a	n/a	n/a	20
Rank value	n/a	n/a	n/a	n/a
Classification algorithm				
J48	25	51.838	66.1765	73.162
BFTree	22.4265	53.309	86.0294	83.088
FT	27.5735	64.706	90.8088	97.427
LMT	24.2647	72.427	93.0147	96.324
RandomForest	31.6176	64.338	74.2647	90.441
RandomTree	22.0588	47.059	57.3529	69.118
REPTree	26.1029	52.941	79.4118	84.927
DecisionTable	25	34.559	54.4118	67.279
FURIA	24.6324	57.721	73.1618	80.515
JRip	18.3824	48.897	45.2206	74.265
NNge	24.2647	58.824	91.1765	92.647
OneR	25	19.853	51.1029	55.147
PART	22.4265	52.206	68.0147	73.529
NaiveBayes	28.3088	65.441	79.0441	95.221
Bagging	29.4118	61.397	87.8676	89.338
MultilayerPerceptron	20.2206	63.971	94.1176	91.912
SMO	20.9559	69.853	93.3824	93.015
Q_1	22.4265	51.838	66.1765	73.529
Min	18.3824	19.853	45.2206	55.147
Median	24.6324	57.721	79.0441	84.927
Mean	24.5675	55.255	75.562282	82.785
Max	31.6176	72.427	94.1176	97.427
Q_3	26.1029	64.338	90.8088	92.647

(c)

LR accuracy (%)	Wavelet	LPC2CC	SFX	SFX + FS
Feature selection method	n/a	n/a	n/a	Cfs
Original no. of attributes	50	10	75	75
No. of attributes after FS	n/a	n/a	n/a	32
Classification algorithm				
J48	80.6667	66	94	94
BFTree	86	71.333	93.3333	93.333
FT	71.3333	66.667	70.6667	72.667
LMT	74.6667	66.667	86.6667	86.667
NBTree	85.3333	69.333	66.6667	67.333
RandomForest	84.6667	68.667	66.6667	90.667
RandomTree	77.3333	66.667	70	85.333
REPTree	86.6667	74	94.6667	94.667
ConjunctiveRule	66	65.333	66.6667	66.667
DecisionTable	82	68.667	93.3333	93.333
FURIA	80.6667	67.333	84	84
JRip	86	68	98	98
NNge	86	68	84.6667	87.333
OneR	84	68	93.3333	93.333
PART	85.3333	66.667	94	94
NaiveBayes	66	74.667	76	78.667
Bagging	86.6667	69.333	93.3333	93.333
LibSVM	71.3333	67.333	50.3333	57.333
MultilayerPerceptron	69.3333	66.667	95.3333	100
SMO	64.6667	66.667	92.6667	98
Q_1	71.3333	66.667	70.500025	82.667
Min	64.6667	65.333	50.3333	57.333
Median	81.3334	67.667	89.6667	92
Mean	78.7333	68.3	83.21667	86.433
Max	86.6667	74.667	98	100
Q_3	85.5	68.833	93.499975	94

(d)

Figure 12: (a) FM boxplot and accuracy table. (b) ES boxplot and accuracy table. (c) SI boxplot and accuracy table. (d) LR boxplot and accuracy table.

5. Results and Analysis

The objective of our experiments is to compare the performance of those four preprocessing methods on four kinds of voice datasets when a collection of data mining classifiers are applied. Our performance evaluation covers four main aspects: (1) accuracy comparison of datasets; (2) accuracy comparison of preprocessing methods; and (3) overall averaged performance comparison.

Twenty popular classification algorithms were used on FM and LR datasets, which is regarded as a representative set of commonly used classifiers. However, the classifier of LibSVM could not be applied on ES and SI due to their formats. Some attribute data contain infinitely small values. Results from some classifiers are not available because of the time limitation: it takes too much time for them to build a classification model when the number of attributes gets very large. As such, LibSVM is excluded from experiments involving ES and SI. NBTree and Conjunctive Rule are excluded from experiments over the dataset SI. For feature selection, the algorithm candidate that yields the highest accuracy is used in the subsequent experiments.

5.1. Accuracy Comparison of Datasets. The accuracy of the classification result is the most significant criterion for evaluating the performance. It is defined as the percentage of correctly classified instances over the total number of instances. This section shows total accuracies of four preprocessing methods on each voice dataset. Four sets of accuracy results and box plots for different dataset are presented in Figures 12(a) to 12(d).

From the aforementioned figures we find that the first two preprocessing methods, which are wavelet and LPC-to-CC, yielded a relatively nonstationary accuracy result on all four datasets. For LR dataset, wavelet method generated better result than LPC-to-CC. Conversely, LPC-to-CC was better for FM, ES, and SI. Recalling from Section 4.1.1, we know that only the LR dataset is synthetic, which was produced by a Text-to-Speech engine. LPC-to-CC, known as a common voice encoding method, has a problem in obtaining the more

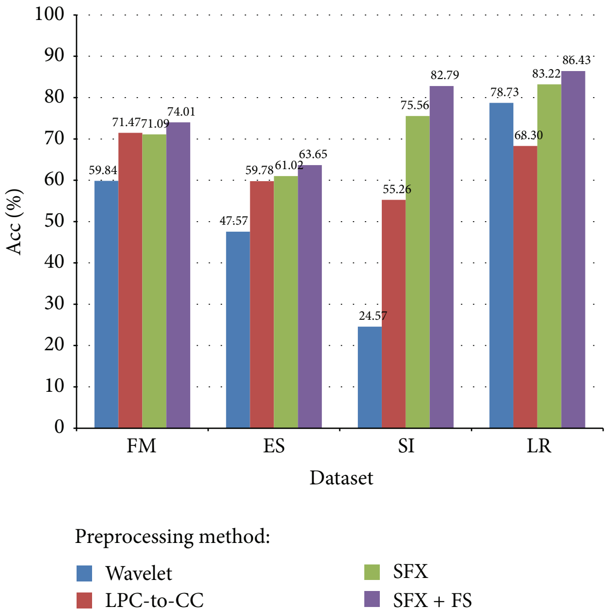


Figure 13: Comparison of average accuracy for different voice datasets and different preprocessing methods.

realistic components: there are many transition frames that the LPC model fails to sort correctly [38]. Such inaccuracy of the model might be due to annoying artifacts like buzzes and tonal noises. So the performance was relatively worse.

Meanwhile, SFX and SFX + FS showed relatively more stable results than the first two. They really improved the accuracy a lot. By a contrast of SFX and SFX + FS, after feature selection, the main range (Q_3–Q_1) of accuracy distribution became narrower and the accuracy results increased.

More evident comparison result is given when the accuracies are averaged out and placed together side by side in a bar chart in Figure 13.

An interesting phenomenon is observed from Figure 13—the accuracy fell a little after SFX compared to LPC-to-CC over FM dataset. However, from the methodology of SFX, we know that cepstral coefficients are involved in the attributes of SFX. This indicates that the classification accuracy may decrease when the number of attributes increases due to the redundancy of those unnecessary features [39]. FM has only binary classes; the performances of the preprocessing methods differ very little compared to those in other datasets that have multiple classes. In particular, SI has 16 different classes; the differences of performance between the preprocessing methods become obvious.

Another considerable fact is also derived from Figure 12 on LR dataset—Wavelet seemed to have a better performance than what LPC-to-CC did. Besides the drawback of LPC encoding method, we can also consider other reasons. The inherent frequency of one's speech is an important acoustic feature for identifying different individuals. Other necessary features may include behavioral patterns (such as voice pitch and speaking style) and human anatomy patterns (like the shape of throat). Remember that the result of LPC-to-CC only contains 10 cepstral coefficients, and the number of target groups to be classified is 16. It contains too few information for correct classification and wavelet provides relatively sufficient features.

Considering the number of classes in each dataset together with the accuracy result, we can find that the accuracy of binary targets classification (FM) is higher than multiple targets classification (ES) and (SI) for the frequency-domain encoding methods. For the time-domain methods like SFX and SFX-FS, good accuracy still can be attained in multiclass classification as in SI where the frequency-domain methods underperform.

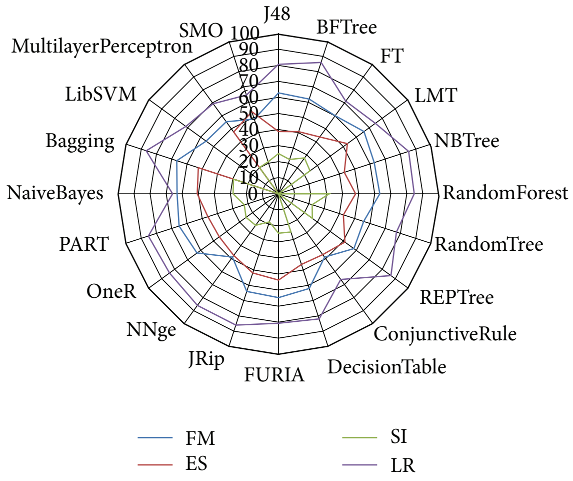
Multiclass classification categorizes instances into more than two classes, whereby a hypothesis is constructed to make sure that discriminates can be distinguished between a fixed set of classes. An assumption is made before that, which is closed set and good distribution. If all possible instances belonging to each case fall into one of the classe, and each class contains statistically representative instances, then the performance of classification is good enough. For now, the boundary of every emotion in ES dataset is not clear (which is already shown in Figure 11(b)), so it does not meet the condition of closed set, and the result is worse than FM. For SI and LR, the features of each individual and language are discriminative enough to tell all classes apart, meaning that they are well distributed, so the results are better than FM.

5.2. Accuracy Comparison of Preprocessing Methods. This section shows the accuracies of four datasets when every preprocessing method is applied on them, respectively. Four sets of accuracy results and radar charts by different preprocessing methods are shown in Figures 14(a) to 14(d).

It can be seen that in general the classification algorithms produce consistent results when wavelet and LPC-to-CC preprocessing methods are used. These almost all-rounded accuracy results are displayed in Figures 14(a) and 14(b). Comparatively, SFX and SFX + FS yield a jagged outline for the curves of accuracy results in the radar chart, which can be seen in Figures 14(c) and 14(d). Overall, Wavelet and LPC-to-CC show lower average accuracy than those in SFX and SFX + FS. Some classifiers produce exceptionally perfect accuracy on all the four datasets after statistical feature extraction and feature selection are applied. They are LMT and Multilayer Perceptron.

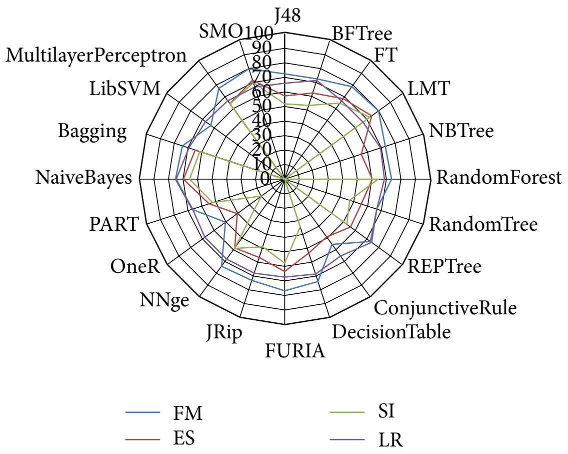
The classifier model generated from LMT is a single tree with different shapes on basis of various types of training data. If the data type is numeric, then a binary tree will be built with splits on those attributes; if the type is nominal, then a multi-split tree is the consequence. But the same thing is that the leaves are each logistic regression model which is quite capable for analysis of dataset with dependent features and bounded magnitudes of time-series. The algorithm is guaranteed that only relevant attributes are selected [40]. The result is much more intelligible and reasonable than a committee of multiple trees on voice classification. So under such kind of circumstance, LMT offers a better result than other tree classifiers.

Multilayer Perceptron is a standard algorithm for any supervised learning task in data mining. The result is



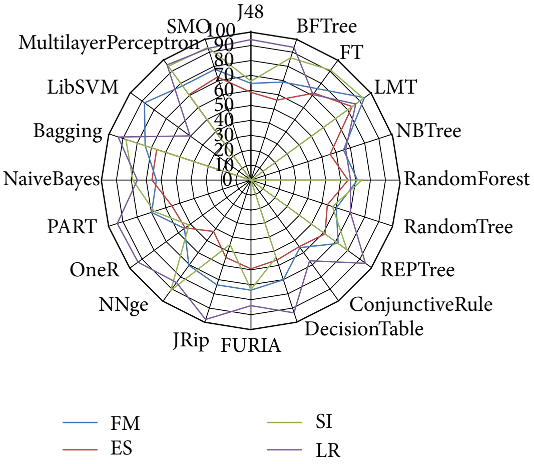
Wavelet	FM	ES	SI	LR
Classification algorithm				
J48	62.7907	38.75	25	80.6667
BFTree	62.0155	40.625	22.4265	86
FT	60.0775	43.75	27.5735	71.3333
LMT	65.8915	53.125	24.2647	74.6667
NBTree	62.7907	43.125	n/a	85.3333
RandomForest	63.1783	48.125	31.6176	84.6667
RandomTree	55.814	42.5	22.0588	77.3333
REPTree	58.1395	51.25	26.1029	86.6667
ConjunctiveRule	48.8372	46.875	n/a	66
DecisionTable	62.0155	46.25	25	82
FURIA	64.7287	53.75	24.6324	80.6667
JRip	63.9535	51.875	18.3824	86
NNge	48.8372	47.5	24.2647	86
OneR	62.7907	45.625	25	84
PART	65.1163	46.25	22.4265	85.3333
NaiveBayes	63.1783	50.625	28.3088	66
Bagging	66.6667	52.5	29.4118	86.6667
LibSVM	55.814	n/a	n/a	71.3333
MultilayerPerceptron	55.4264	47.5	20.2206	69.3333
SMO	48.8372	53.75	20.9559	64.6667

(a)



LPC2CC	FM	ES	SI	LR
Classification algorithm				
J48	72.4806	57.25	51.8382	66
BFTree	72.4806	62.25	53.3088	71.3333
FT	78.6822	68.5	64.7059	66.6667
LMT	79.845	74.125	72.4265	66.6667
NBTree	72.8682	55.375	n/a	69.3333
RandomForest	73.6434	60.375	64.3382	68.6667
RandomTree	66.6667	55.375	47.0588	66.6667
REPTree	72.8682	56	52.9412	74
ConjunctiveRule	55.4264	49.125	n/a	65.3333
DecisionTable	74.031	53.5	34.5588	68.6667
FURIA	76.7442	63.5	57.7206	67.3333
JRip	73.2558	56.625	48.8971	68
NNge	74.031	58.5	58.8235	68
OneR	50.3876	40.375	19.8529	68
PART	67.0543	54.75	52.2059	66.6667
NaiveBayes	75.1938	69.75	65.4412	74.6667
Bagging	74.031	64.75	61.3971	69.3333
LibSVM	62.7907	n/a	n/a	67.3333
MultilayerPerceptron	77.1318	64.125	63.9706	66.6667
SMO	79.845	71.625	69.8529	66.6667

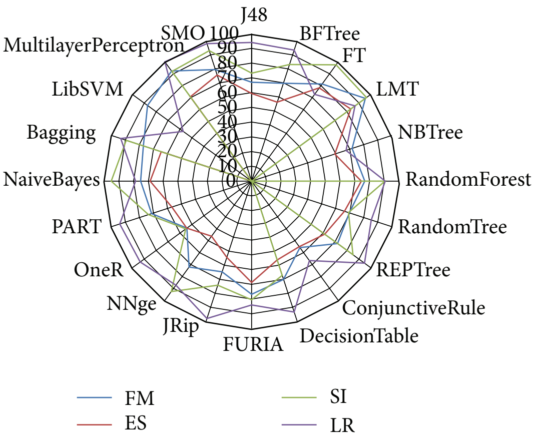
(b)



SFX	FM	ES	SI	LR
Classification algorithm				
J48	64.7287	58.75	66.1765	94
BFTree	68.9922	56.25	86.0294	93.3333
FT	75.5814	71.875	90.8088	70.6667
LMT	93.7984	83.75	93.0147	86.6667
NBTree	65.1163	55.625	n/a	66.6667
RandomForest	71.3178	65	74.2647	66.6667
RandomTree	60.0775	53.75	57.3529	70
REPTree	72.093	61.25	79.4118	94.6667
ConjunctiveRule	55.814	55	n/a	66.6667
DecisionTable	70.155	56.25	54.4118	93.3333
FURIA	73.6434	59.375	73.1618	84
JRip	73.6434	54.375	45.2206	98
NNge	70.5426	42.5	91.1765	84.6667
OneR	55.0388	54.375	51.1029	93.3333
PART	69.7674	55.625	68.0147	94
NaiveBayes	63.1783	66.25	79.0441	76
Bagging	74.4186	66.25	87.8676	93.3333
LibSVM	87.9845	n/a	n/a	50.3333
MultilayerPerceptron	77.907	70.625	94.1176	95.3333
SMO	77.907	72.5	93.3824	92.6667

(c)

Figure 14: Continued.



SFX + FS	FM	ES	SI	LR
Classification algorithm				
J48	67.0543	59.375	73.1618	94
BFTree	69.7674	56.25	83.0882	93.3333
FT	81.3953	78.125	97.4265	72.6667
LMT	95.3488	83.125	96.3235	86.6667
NBTree	71.7054	59.375	n/a	67.3333
RandomForest	76.7442	74.375	90.4412	90.6667
RandomTree	69.7674	66.25	69.1176	85.3333
REPTree	72.093	61.25	84.9265	94.6667
ConjunctiveRule	55.814	55	n/a	66.6667
DecisionTable	70.155	56.25	67.2794	93.3333
FURIA	76.7442	68.75	80.5147	84
JRip	64.7287	54.375	74.2647	98
NNge	72.093	46.25	92.6471	87.3333
OneR	54.2636	54.375	55.1471	93.3333
PART	71.3178	56.875	73.5294	94
NaiveBayes	75.1938	68.75	95.2206	78.6667
Bagging	77.1318	64.375	89.3382	93.3333
LibSVM	87.2093	n/a	n/a	57.3333
MultilayerPerceptron	92.2481	70.625	91.9118	100
SMO	79.4574	75.625	93.0147	98

(d)

Figure 14: (a) Accuracy comparison of Wavelet preprocessing method. (b) Accuracy comparison of LPC-to-CC preprocessing method. (c) Accuracy comparison of SFX preprocessing method. (d) Accuracy comparison of SFX + FS preprocessing method.

relatively better than any other classifiers, achieving almost 100% accuracy but the time cost is higher and sometimes unacceptable. However, some classifiers produce low accuracy, for instance, Naïve Bayes. Based on Bayes' theorem with strong independence assumptions, Naïve Bayes acts as quite a simple classifier and it gets very widely adopted in many classification situations. But sometimes the relation between any pair of attributes is always dependent and the distribution of features is unknown in advance; thus the performance of such a simple probabilistic classifier is bad and unstable.

5.3. Overall Averaged Performance Comparison.

5.3. Overall Averaged Performance Comparison. For a throughout performance evaluation, performance consideration of other parameters is considered as well; these include Kappa, Precision, Recall, F1, and ROC, which are commonly used in assessing the quality of the classification models in data mining. These performance indicators are briefly described as follows. The performance results pertaining to these indicators are averaged over all the four datasets and all the 20 classification algorithms. They are then shown in Section 5.3.6 together with the comparison of time cost.

5.3.1. Kappa Statistic. Kappa statistic is widely used to measure variability between multiple observers. The meaning of Kappa statistic is how often multiobservers agree in terms of their interpretations. When two or more evaluators are checking the same data, Kappa statistic is assessed to show an agreement of evaluators when the same data categories are correctly assigned. As well known, simple agreement just between yes and no is poor because of the property of chance and arbitrary. That is why Kappa statistic is introduced and it is preferred [41]. The definition of Kappa statistic is given as follows:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}, \qquad (34)$$

where $\Pr(a)$ is the relative observed agreement among raters and $\Pr(e)$ is the hypothetical probability of chance agreement. When the application is classification, the measure of chance between the classification results and the true classes (labeled categorical data class) is assessed by Kappa statistic. It reflects the reliability of the evaluation of our classifier. Table 6 is the general criterion of evaluating Kappa statistic [42]. A comparison of different voice datasets and different preprocessing methods, in terms of average Kappa statistic, is shown in Figure 15. Wavelet method is relatively unstable in datasets of FM, ES, and SI. The Kappa statistics for LPC-CC method are almost the same across different datasets. SFX without FS, however, underperformed when compared to LPC-CC in FM and ES datasets which are relatively simple. SFX-FS shows its superiority in Kappa statistics in all datasets.

5.3.2. Precision. In pattern recognition and data mining, precision is the fraction of relevantly retrieved instances. In the situation of classifications, the terms positive and negative describe the classifier's prediction results, and the terms true and false refer to whether the prediction results correspond to the fact or not [43]. This is illustrated by Table 7.

Precision is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \qquad (35)$$

Precision is concisely defined as "of all the instances that were classified into a particular class, how many were actually belonged to that class?" In classification task, a perfect precision score for a particular class means that every instance classified into that class does indeed belong to that class (but it says nothing about the number of instances from that class that were not classified correctly). As shown in Figure 16, for example, SFX-FS when applied on LR dataset has the maximum precision score 0.88—that means 88% of

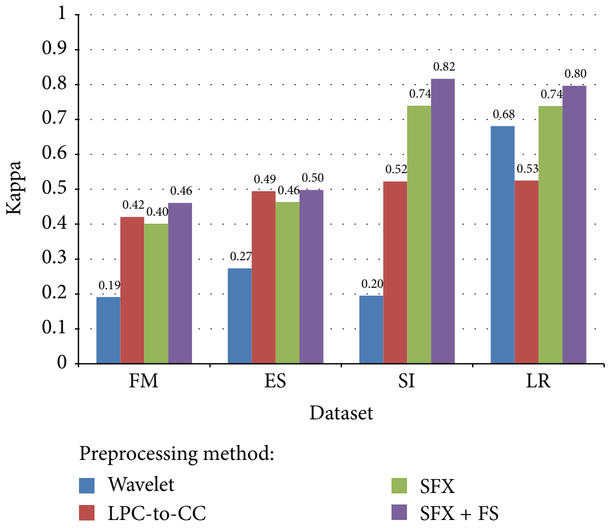


Figure 15: Comparison of average Kappa statistic for different voice datasets and different preprocessing methods.

Table 6: Strength of agreement of Kappa statistic.

Kappa	Agreement	Interpretation
<0	Less than chance agreement	Poor
0.01–0.20	Slight agreement	Slight
0.21–0.40	Fair agreement	Fair
0.41–0.60	Moderate agreement	Moderate
0.61–0.80	Substantial agreement	Substantial
0.81–1.00	Almost perfect agreement	Almost perfect

the instances that are classified into a particular indeed belong to that class. SFX-FS for SI has precision score 0.85, for ES has only 0.64, and for FM has 0.73. Wavelet method was unacceptable for all datasets except LR, for it has merely 0.59, 0.42, and 0.25 precision scores, respectively. The comparison with respect to precision scores is shown in Figure 16.

5.3.3. Recall. In pattern recognition and data mining, recall is defined as the fraction of relevantly retrieved instances. We can infer that the same part of both precision and recall is relevance, based on which they all make a measurement. Usually, precision and recall scores are not discussed in isolation and the relationship between them is inverse, indicating that one increases and the other decreases. Recall is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{36}$$

In a classification task, recall is a criterion of the classification ability of a prediction model to select labeled instances from training and testing datasets. A recall of score 1.0 means that each instance from that particular class is labeled to this class and all are predicted correctly, and none shall be left out [44]. Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e., the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been).

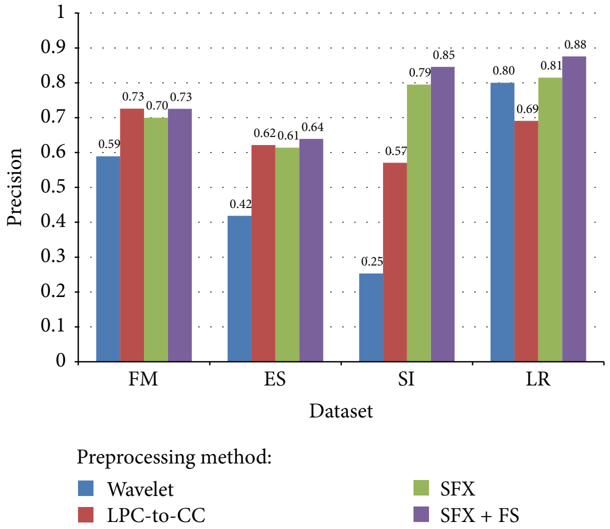


Figure 16: Comparison of average precision for different voice datasets and different preprocessing methods.

Table 7: Definitions of precision and recall terms.

	Actual Class (Observation)	
Predicted Class (Expectation)	TP (True Positive) Correct Result	FP (False Positive) Unexpected Result
	FN (False Negative) Missing Result	TN (True Negative) Correct Absence of Result

The recall scores defined loosely as "of all the instances that are truly of a particular class, how many did we classify them into that class?" For example, as shown in Figure 17, 86% of instances are classified into the classes and they actually belonged to those classes. Inversely 14% is missed out. Again, the recall scores for Wavelet method are comparatively low except in the LR dataset it exceeds that of LPC-to-CC method. Having a low recall score means the classifier is conservative. SFX-FS is outperforming the rest of the methods in terms of recall scores. The comparison is shown in Figure 17.

5.3.4. F-Measure. F-measure is the harmonic mean of precision and recall, that is,

$$\begin{aligned} F\text{-measure} &= \frac{2}{1/\text{Precision} + 1/\text{Recall}} \\ &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{37}$$

It is also known as balanced F score or F-measure in tradition, because recall and precision are equally weighted. The general formula for F_β measure is

$$\begin{aligned} F_\beta &= \frac{1 + \beta^2}{1/\text{Precision} + \beta^2/\text{Recall}} \\ &= \frac{\left(1 + \beta^2\right) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}. \end{aligned} \tag{38}$$

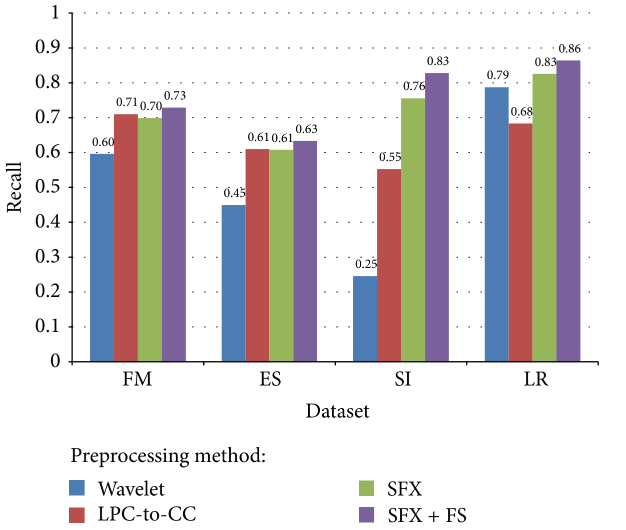


Figure 17: Comparison of average recall for different voice datasets and different preprocessing methods.

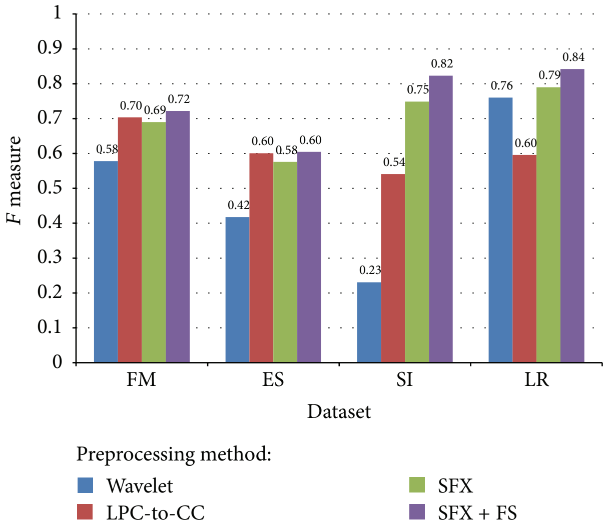


Figure 18: Comparison of average F-measure for different voice datasets and different preprocessing methods.

Table 8: Overall Averaged Performance Comparison of Pre-processing Methods.

Average performance	Pre-processing methods			
	Wavelet	LPC-2-CC	SFX	SFX + FS
Accuracy %	52.67789	63.70274	72.72099	**76.72044**
Kappa Statistics	0.335301	0.490773	0.58568	**0.643008**
Precision	0.515225	0.652195	0.730832	**0.771412**
Recall	0.519617	0.638896	0.721978	**0.763601**
F-measure	0.496758	0.610196	0.701144	**0.747919**
ROC	0.717222	0.787528	0.836521	**0.859025**

As mentioned before, precision and recall scores should be taken into account simultaneously because they have a strong relation essentially. Consequentially, both are combined into a single measure, which is F-measure. Other complicated combinations of precision and recall include but are not limited to the weighted harmonic mean of precision and recall (F_β), and the geometric mean of regression coefficients, and Informedness and Markedness (Matthews correlation coefficient [45]). In our experiments, we only concern F_1-measure. F_1 measure is a derived effectiveness measurement. The resultant value is interpreted as a weighted average of the precision and recall. The best value is 1 and the worst is 0. Figure 18 shows a comparison of average F_1 measure for different voice datasets and different preprocessing methods. SFX-FS shows superior F_1 score in datasets SI and LR; it duels with LPC-to-CC in simple datasets like FM and ES.

5.3.5. ROC. A Receiver Operating Characteristic (ROC) is generated by plotting True Positive Rate (TPR) verse False Positive Rate (FPR) with many value settings of threshold. It is a graphical plot which illustrates the performance of sensitivity and specificity. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate. A ROC space is defined by FPR and TPR as x and y axes, respectively, with the coordinate $(0, 1)$ representing the best prediction result. The area-under-curve (AUC) statistic of ROC is commonly used in machine learning and data mining community for model comparison. The AUC is an equivalent and simple replacement of ROC curve.

ROC is useful for gaining insight into the decision-making ability of the model—how likely is the classification model to accurately predict the respective classes? The AUC measures the discriminating ability of a classification model. The larger the AUC, the higher the likelihood that an actual positive case will be assigned a higher probability of being positive than an actual negative case. The AUC measure is especially useful for datasets with unbalanced target distribution (one target class dominates the other). A comparison in terms of ROC AUC which is normalized to $[0, 1]$ for different voice datasets and different preprocessing methods is shown in Figure 19. Again, they show similar performance results to those in F_1 measures. SFX + FS perform equally well in SI dataset and LR dataset with 0.94 AUC; it is slightly higher than SFX and LPC-to-CC in FM and ES datasets. Wavelet has the lowest AUC in all datasets except LR where it is better than that of LPC-to-CC.

5.3.6. Aggregated Results. The final results that are averaged and aggregated, from the individual results tested by using different datasets and different classification algorithms, are shown as follows. We compare in particular various preprocessing methods against a collection of performance indicators, as in Table 8.

From Table 8, we can reach a conclusion that SFX with FS is indeed the most suitable preprocessing method for all types of voice datasets. It has a higher value across all performance indicators than the rest of the preprocessing methods.

The accuracy and CPU time are evaluated across different feature selection algorithms; the averaged results together

Table 9: Overall averaged performance comparison of ensemble feature selections.

FS	No. attributes from frequency domain	No. attributes from time domain	Total no. attributes	No. attributes after FS	Average CPU time (s)	Av. Acc. %
CFS	10	66	76	19	1.28	74.25
ChiSq	10	66	76	52	3.05	73.44
MRMR	10	66	76	30	3.26	68.54
WSA	10	66	76	25	1240 (min. 31)	75.29

Table 10: Overall averaged time cost comparison.

Time	Preprocessing				FS		Build Model	Total
Dataset	LPC2CC	DS	DTW	Piecewise				
FM	10 s	5 m 23 s	15 m 3 s	32 m	CFS	0.78 s	1.13 s	52 m 37.9 s
					ChiSq	2.867 s		52 m 40 s
					MRMR	3.56 s		52 m 40.7 s
					WSA	31.275 s		53 m 18.4 s
ES	9.5 s	9 m 35 s	21 m 38 s	1 h 13 m	CFS	1.03 s	1.25 s	1 h 44 m 24.8 s
					ChiSq	3.328 s		1 h 44 m 27.1 s
					MRMR	1.439 s		1 h 44 m 25.2 s
					WSA	441.476 s		1 h 51 m 45.2 s
SI	15.8 s	25 m 6 s	38 m 23 s	2 h 14 m	CFS	1.91 s	1.7 s	3 h 17 m 48.4 s
					ChiSq	3.815 s		3 h 17 m 50.3 s
					MRMR	3.26 s		3 h 17 m 49.8 s
					WSA	3585 s		4 h 17 m 31.5 s
LR	13.4 s	16 m 48 s	42 m 45 s	1 h 57 m	CFS	1.39 s	1.56 s	2 h 56 m 49.4 s
					ChiSq	2.17 s		2 h 56 m 50.1 s
					MRMR	4.8 s		2 h 56 m 52.8 s
					WSA	906 s		3 h 11 m 54 s

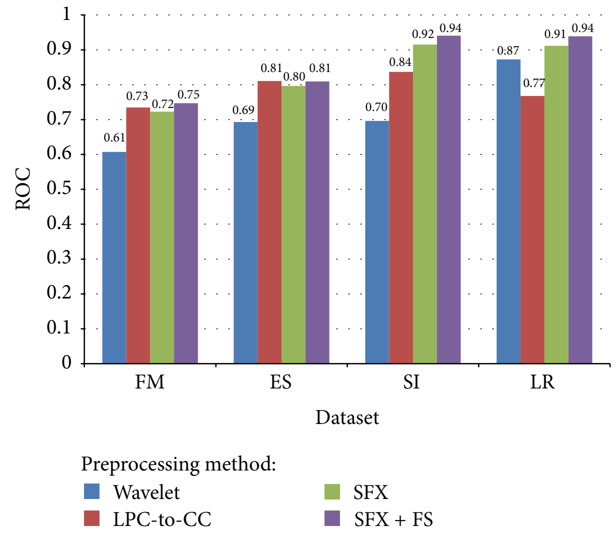


Figure 19: Comparison of average ROC AUC for different voice datasets and different preprocessing methods.

with the amount of attributes before and after FS are shown in Table 9.

In Table 9, the first three FS algorithms have been widely used, and the last one is recently proposed by Fong [20]. WSA gives the second fewest number of attributes after feature selection, highest classification accuracy, and a compromising time cost with 31 seconds minimum. So to some extent WSA is a good choice of feature selection if time requirement is not a concern in training up a voice classification model. WSA is done at the cost of incurring extra time in doing the heuristic optimization on the feature subset.

Table 9 shows the overall averaged time cost of each process step applied on different datasets. Piecewise transformation and DTW need much longer time than the other processes due to the computational complexity. The time consumption by piecewise transformation is relatively long especially for complex datasets like SI and LR. Statistic measures are computed for each segment (20x) for each time-series. WSA works as a stochastic iteration model, which progressively refines the performance and is superior to the other three FS methods but comes at a certain time cost. In contrast the classification model construction times in general are very short, with an average of less than two seconds. Please see Table 10. The total time required for preprocessing voice data for classification ranges from slightly less than an hour to four hours and eighteen minutes, depending on the choice of preprocessing algorithms and complexity of the datasets. Be reminded that the reference of time consumption shown here is for training a classifier based on the given training set; once a classifier is trained, the testing is very fast that takes almost no time. Therefore, a system designer can choose the best performing algorithms in terms of accuracy and other performance quality indicators if the voice classification application is not prone to frequent update of training dataset (that means no need to build the classification model over again), and of course vice versa this implies.

6. Conclusion and Future Works

Human voice is referred to as one of the bodily vital signs that could be measured, recorded, and analyzed as fluctuations of amplitude of sound loudness. Voice classification constitutes to a number of biometrics techniques of which the theories have been formulated, studied, and implemented in practical applications. Traditional classification algorithms from data mining domain, however, require the input of training data to be formatted in a data matrix where the columns represent features/attributes that characterize the voice data, and the rows are the instances of the voice data. Each record must have a verdict known as predicted class for training data. In the literature, mainly the characteristics of voice data are acquired from the frequency domain, for example, LPC, cepstral coefficients, and MFCC. Those popular preprocessing methods have demonstrated significant advantages in transforming voice data which is in the form of time-series to signatures in the frequency domain. There exist possibilities that some useful attributes can be harvested from the time domain considering the temporal patterns of voice data that are supposedly distinctive from one another. A challenge to overcome is its expensive computational cost of time and large search space in the time domain.

Considering the stochastic and nonstationary nature of human voice, a hybrid data preprocessing methodology is adopted in voice classification in this paper, where combined analysis from both frequency and time domain is included. In particular, a time domain feature extraction technique called Statistics Feature Extraction (SFX) is presented. SFX utilizes piecewise transformation that partitions a whole time-series into segments and statistics features are extracted subsequently from each piece. Simulation experiments were conducted on classifying four types of voice data, namely, Female and Male, Emotional Speech, Speaker Identification, and Language Recognition into different groups by using SFX and its counterparts (SFX and Feature Selection). The results showed that SFX is able to achieve a higher accuracy in the classification models for the four types of voice data.

The contribution is significant as the new preprocessing methodology can be adopted by fellow researchers that will enable them to build more accurate voice classification model. Besides, the feature selection result proves that a metaheuristic feature selection algorithm called Wolf Search (WSA) can achieve a global optimal feature subset for highest possible classification accuracy. As there is no free lunch in the world, WSA costs considerable amount of computational time.

The precision of piecewise transformation segmentation can be one of the future works. If the number of segments is too large (low resolution in time-series modeling), then it will lead to the low accuracy of feature extraction; if the window is too small (with very refined resolution), then a lot more computational costs are incurred. Although calibration was done beforehand for calculating the ideal segment length for subsequent processing, this again contributes to extra processing time, and the calibrated result may need to be refreshed should the natures of the voice data evolve. Some dynamic and incremental methods are opted for solving this calibration problem for estimating the correct length of segments. Furthermore the segment lengths can be variables that cope with the level of fluctuation of the voice data, dynamically.

Acknowledgments

The authors are thankful for the financial support from the research grant “Adaptive OVFDT with Incremental Pruning and ROC Corrective Learning for Data Stream Mining,” Grant no. MYRG073(Y2-L2)-FST12-FCC, offered by the University of Macau, FST, and RDAO.

References

[1] S. Fong, “Using hierarchical time series clustering algorithm and wavelet classifier for biometric voice classification,” *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 215019, 12 pages, 2012.

[2] S. Fong, K. Lan, P. Sun, O. Mohammed, J. Fiaidhi, and S. Mohammed, “A timeseries pre-processing methodology for biosignal classification using statistical feature extraction,” in *Proceedings of the 10th IASTED International Conference on Biomedical Engineering (Biomed ’13)*, pp. 207–214, Innsbruck, Austria, February 2013.

[3] C. F. Chan and W. M. E. Yu, “An abnormal sound detection and classification system for surveillance applications,” in *Proceedings of the European Signal Processing Conference (EUSIPCO ’10)*, pp. 1–2, Aalborg, Denmark, August 2010.

[4] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the cuidado project,” CUIDADO Project Report, 2004.

[5] C. Aguiar, *Modelling the Excitation Function to Improve Quality in LPC’s Resynthesis*, Center for Computer Research in Music and Acoustics, Stanford University, Stanford, Calif, USA.

[6] L. R. Rabiner and M. R. Sambur, “Application of an LPC distance measure to the voiced-unvoiced-silence detection problem,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 338–343, 1977.

[7] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, NJ, USA.

[8] G. Antoniol, V. F. Rollo, and G. Venturi, “Linear Predictive Coding and Cepstrum coefficients for mining time variant information from software repositories,” in *Proceedings of the 2005 International Workshop on Mining Software Repositories*, pp. 1–5, July 2005.

[9] N. Awasthy, J. P. Saini, and D. S. Chauhan, “Spectral analysis of speech: a new technique,” *International Journal of Information and Communication Engineering*, vol. 2, no. 1, pp. 19–28, 2006.

[10] B. Logan, “Mel frequency cepstral coefficients for music modeling,” in *Proceedings of the International Symposium on Music Information Retrieval*, pp. 1–3, 2000.

[11] S. V. Chapaneri, “Spoken digits recognition using weighted MFCC and improved features for dynamic time warping,”

International Journal of Computer Applications, vol. 40, no. 3, pp. 6–12, 2012.

[12] X. Zhou, Y. Fu, M. Liu, M. Hasegawa-Johnson, and T. S. Huang, "Robust analysis and weighting on MFCC components for speech recognition and speaker identification," in *Proceedings of the IEEE International Conference onMultimedia and Expo (ICME '07)*, pp. 188–191, July 2007.

[13] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[14] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[15] T. Nitta, "Feature extraction for speech recognition based on orthogonal acoustic-feature planes and LDA," in *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-99)*, pp. 421–424, March 1999.

[16] J. H. Lee, H. Y. Jung, T. W. Lee, and S. Y. Lee, "Speech feature extraction using independent component analysis," in *Proceedings of the IEEE Interntional Conference on Acoustics, Speech, and Signal Processing*, pp. 1631–1634, June 2000.

[17] B. J. Lee, B. Ku, K. Park, K. H. Kim, and J. Y. Kim, "A new method of diagnosing constitutional types based on vocal and facial features for personalized medicine," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 818607, 8 pages, 2012.

[18] D. Maunder, J. Epps, E. Ambikairajah, and B. Celler, "Robust sounds of activities of daily living classification in two-channel audio-based telemonitoring," *International Journal of Telemedicine and Applications*, vol. 2013, Article ID 696813, 12 pages, 2013.

[19] K. Chenausky, J. MacAuslan, and R. Goldhor, "Acoustic analysis of PD speech," *Parkinson's Disease*, vol. 2011, Article ID 435232, 13 pages, 2011.

[20] S. Fong, "Opportunities and challenges of integrating bio-inspired optimization and data mining algorithms," in *Swarm Intelligence and Bioinspired Computation*, chapter 18, pp. 385–401, Elsevier, 2013.

[21] R. Daniloff, G. Schuckers, and L. Feth, *The Physiology of Speech and Hearing: An Introduction*, Prentice Hall, 1980.

[22] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1975.

[23] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[24] A. Ó. Cinnéide, *Linear Prediction: The Technique, Its Solution and Application to Speech*, Dublin Institute of Technology, Dublin, Ireland.

[25] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[26] G. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, Prentice-Hall, 3rd edition, 1994.

[27] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[28] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[29] VidTIMIT Audio-Video Dataset, *Conrad Sanderson, 2001–2009*, School of Information Technology and Electrical Engineering (ITEE), University of Queensland, St Lucia, Australia, 2013, http://itee.uq.edu.au/~conrad/vidtimit/.

[30] "A database of German emotional speech," Institute of Communication Science of the TU-Berlin (Technical University of Berlin) and funded by the German Research Community (DFG), 2013, http://pascal.kgw.tu-berlin.de/emodb/.

[31] Y. Obuchi, The PDA speech database, Carnegie Mellon University (CMU), 2003, http://www.speech.cs.cmu.edu/databases/pda/README.html.

[32] Microsoft Text-to-Speech engine, 2013, http://msdn.microsoft.com/en-us/library/hh361572.aspx.

[33] R. Tang and S. Fong, "Wolf search algorithm with ephemeral memory," in *Proceedings of the 7th International Conference on Digital Information Management (ICDIM '12)*, pp. 1–3, University of Macau, Macau, China, August 2012.

[34] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK, 2008.

[35] "Chi2 Feature Selection," Stanford Natural Language Processing Group, 2009, http://nlp.stanford.edu/IR-book/html/htmledition/feature-selectionchi2-feature-selection-1.html.

[36] F. García López, M. García Torres, B. Melián Batista, J. A. Moreno Pérez, and J. M. Moreno-Vega, "Solving feature subset selection problem by a parallel scatter search," *European Journal of Operational Research*, vol. 169, no. 2, pp. 477–489, 2006.

[37] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[38] M. A. Osman, A. Nasser, H. M. Magboub, and S. A. Alfandi, "Speech compression using LPC and wavelet," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET '10)*, vol. 7, pp. 92–99, April 2010.

[39] A. G. Janecek, W. N. Gansterer, M. A. Demel, and G. F. Ecker, "On the relationship between feature selection and classification accuracy," *JMLR Workshop and Conference Proceedings*, vol. 4, pp. 90–105, 2008.

[40] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[41] J. Carletta, "Squibs and discussions: assessing agreement on classification tasks: the kappa statistic," *Computational Linguistics*, vol. 22, no. 2, pp. 248–254, 1996.

[42] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: the kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005.

[43] P. M. W. David, "Evaluation: from precision, recall and F-factor to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[44] D. L. Olson and D. Dursun, *Advanced Data Mining Techniques*, Springer, 1st edition, 2008.

[45] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 656391, 8 pages
http://dx.doi.org/10.1155/2013/656391

Research Article

Statistical Fractal Models Based on GND-PCA and Its Application on Classification of Liver Diseases

Huiyan Jiang,[1] Tianjiao Feng,[1] Di Zhao,[1] Benqiang Yang,[2] Libo Zhang,[2] and Yenwei Chen[3]

[1] *Software College, Northeastern University, Shenyang 110819, China*
[2] *Radioactive Branch, PLA General Hospital of Shenyang Military Region, Shenyang 110016, China*
[3] *Department of Information Science and Engineering, Ritsumeikan University, Shiga 5258577, Japan*

Correspondence should be addressed to Huiyan Jiang; hyjiang@mail.neu.edu.cn

Received 1 July 2013; Accepted 31 August 2013

Academic Editor: Tai-hoon Kim

A new method is proposed to establish the statistical fractal model for liver diseases classification. Firstly, the fractal theory is used to construct the high-order tensor, and then Generalized N-dimensional Principal Component Analysis (GND-PCA) is used to establish the statistical fractal model and select the feature from the region of liver; at the same time different features have different weights, and finally, Support Vector Machine Optimized Ant Colony (ACO-SVM) algorithm is used to establish the classifier for the recognition of liver disease. In order to verify the effectiveness of the proposed method, PCA eigenface method and normal SVM method are chosen as the contrast methods. The experimental results show that the proposed method can reconstruct liver volume better and improve the classification accuracy of liver diseases.

1. Introduction

Liver cancer is a common disease in daily life and is often diagnosed when it is advanced, and very few liver cancer patients can be cured. So it is necessary for us to diagnose liver cancer as early as possible. Computer Aided Diagnosis (CAD) technology is established with the development of computer graphics technology, image processing technology, and pattern recognition technology. Since CT images increase the burden of doctors, research of this kind of technology is urgently needed. In recent years, scientists have researched several typical variable models such as Snake Model. These methods are more suitable for objects with smooth boundary and do not use the valuable prior knowledge. Active Texture Model (ATM) which is evolved from them can reflect the texture feature of the object. The roughness of surface in medical images is an important factor to distinguish among lesions, so ATM cannot represent object features well only with the texture model of gray feature.

In order to solve the problems as above, we proposed the statistical fractal model based on the feature of gray level and fractal dimension. The statistical fractal model can be better used in the analysis of medical images such as diseases recognition, but the construction of statistical appearance model is a challenging task when the number of training samples is much fewer than the number of dimensions of data.

Principal Component Analysis (PCA) method [1] is a famous method used in the subspace recognition, and it is one of the classical methods based on statistical feature. But this method has two problems. The first is that the original space structure of image is damaged in the vectorization process. The second is that it may cause the dimension disaster when we transfer the image into a vector. So we need more large space to calculate the covariance matrix of images. In order to solve these problems, we use the Generalized N-dimensional PCA [2] to learn subspace in this paper.

Support Vector Machine (SVM) [3] is commonly used to train a classifier. And the factor to affect the classification performance is the parameters used in SVM. So we use Ant Colony Optimization (ACO) algorithm to optimize SVM parameters, and then we use Directed Acyclic Graph DAG [4] to multiclassify liver diseases.

As above, for protecting the special space structure information of liver images and solving the dimension disaster

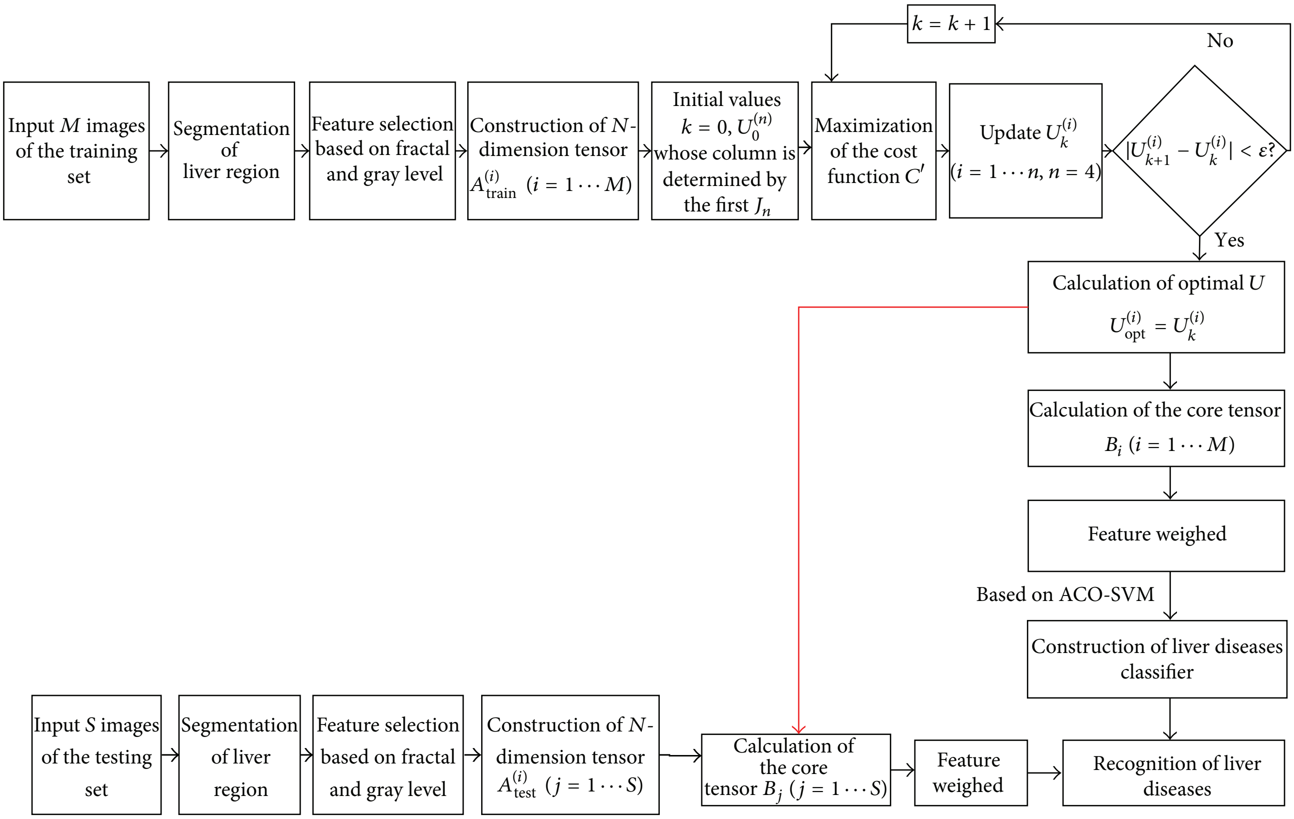

Input M images of the training set
Segmentation of liver region
Feature selection based on fractal and gray level
Construction of N-dimension tensor $A_{\text{train}}^{(i)}$ $(i = 1 \cdots M)$
Initial values $k = 0, U_0^{(n)}$ whose column is determined by the first J_n
Maximization of the cost function C'
Update $U_k^{(i)}$ $(i = 1 \cdots n, n = 4)$
$|U_{k+1}^{(i)} - U_k^{(i)}| < \varepsilon$?
No
$k = k + 1$
Yes
Calculation of optimal U $U_{\text{opt}}^{(i)} = U_k^{(i)}$
Calculation of the core tensor B_i $(i = 1 \cdots M)$
Feature weighed
Based on ACO-SVM
Construction of liver diseases classifier
Input S images of the testing set
Segmentation of liver region
Feature selection based on fractal and gray level
Construction of N-dimension tensor $A_{\text{test}}^{(i)}$ $(j = 1 \cdots S)$
Calculation of the core tensor B_j $(j = 1 \cdots S)$
Feature weighed
Recognition of liver diseases

Figure 1: The main flow of the proposed method.

problem, we extracted the gray feature and fractal feature to establish the high-order tensor of liver volume and constructed the statistical fractal model with GND-PCA method. For improving recognition accuracy, SVM optimized by ACO (ACO-SVM) was used to recognize liver diseases images.

This paper is organized as follows. Section 2 introduces the proposed method; firstly, we will introduce some knowledge of PCA and tensor, then we will show the construction of high-order tensor, and finally we introduce the method of GND-PCA for the construction of statistical fractal model and ACO-SVM [5] for classification. In Section 3, we present the construction of liver images after GND-PCA and the results of classification. Section 4 concludes the works in this paper.

2. Materials and Methods

In this section, we will introduce some background knowledge about GND-PCA method firstly. We mainly present the method of PCA, 2D-PCA, and ND-PCA and the basic knowledge of tensor. And then we will introduce our method of construction of statistical fractal model. The main flow is shown in Figure 1. The process of the proposed method is described as follows.

(1) Liver images preprocessing. Firstly we segment the CT images of abdomen to gain the liver region of image, and then we calculate the fractal dimension of liver image.

(2) High-order tensor construction. At first, we collect a group of fractal features and gray level features (pixels), and then we combine them into a new dataset.

(3) Statistical fractal model establishment. In this paper, we use the method of Generalized N-dimensional PCA (GND-PCA) to establish the statistical fractal model for classification.

(4) Liver diseases classification based on ACO-SVM. After we obtain the statistical fractal model by GND-PCA, we treat the core tensors as samples, and then we use SVM optimized by ACO to classify liver diseases.

2.1. PCA Method and Its Extension. PCA is an application of K-L conversion in statistics. The purpose of PCA is to lower the dimension of data through finding a linear mapping. The mapping meets the following conditions.

(1) The error of sample reconstruction is minimized.

(2) The mapping of sample set in low dimension space has the maximum variance.

(3) The correlation among samples is erased.

Turk and Pentland proposed the famous method named eigenface to realize PCA. Suppose that we have K training

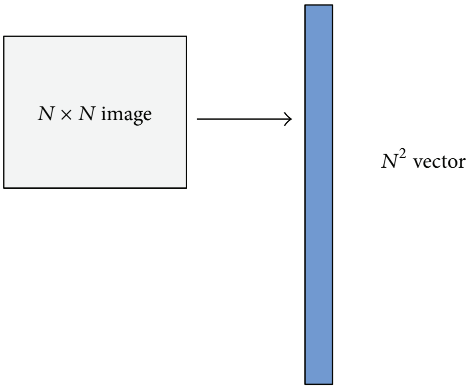


Figure 2: 2D matrix to vector.

samples, $I_1, I_2, \ldots, I_K$. Firstly, we transfer these samples into vectors shown in Figure 2. The $N \times N$ image is transferred into a column vector; that is to say, the training samples $I_1, I_2, \ldots, I_K$ are transferred into $X_1, X_2, \ldots, X_K$. X_i is a column vector with the dimension of n ($n = N^2$). Each X_i is in the space of n dimension. According to the knowledge of linear algebraic, X_i can be expressed by n basis in the n dimension space. If we express $X_1, X_2, \ldots, X_K$ by only one vector, obviously, we should use the average value m of $X_1, X_2, \ldots, X_K$. We rename m as the zero-dimension expression of sample datasets. Doing as above is useful and easy, but the shortcoming of it is that it can not show the difference between samples. So the second step of PCA is to centralize the training sample sets $I_1, I_2, \ldots, I_K$, and then we find the 1-dimension to d dimension expression of the new sample sets.

Compared with PCA, 2D-PCA uses the 2-dimension image matrix directly for feature extraction. It can calculate the covariance matrix accurately with less time. Imagining that there is an n dimension column vector which is normalized, we can project any image ($m \times n$) to it, $Y = AX$, and get the m dimension image eigenvector. The separating capacity of X can be measured by the total divergence of the projected samples, and the total divergence of projected samples can be expressed by the trace of the covariance of the reflected eigenvector. $J(X) = \mathrm{tr}(S_x)$, S_x is the covariance of the projected eigenvector of training sample, and $\mathrm{tr}(S_x)$ is the trace of S_x. The purpose of maximization of $J(X)$ is to find the mapping direction, and the final total divergence of mapping sample is the largest.

The advantage of 2D-PCA method is that we do not need to transform images into vectors, and we can use the images themselves directly to deal with data information and find a group of basis which can express the original samples best. Moreover, the eigenfactor is a matrix not a vector which PCA method needs. It keeps the space structure of the original images. And it does not only wipe off the correlation between the samples effectively but also wipe off the correlation between the rows in one sample. But the method has shortages too, and the mapping coefficient matrix is large and wastes lots of memory space because of the ignorance of the difference between the columns in one sample.

Alternative 2D-PCA is proposed to overcome these problems as above. The method can solve the problem of ignorance of the difference between the columns in one sample but also cannot solve the problem of the large coefficient matrix and the difference between both columns and rows. As a result, the G2D-PCA method is proposed, and this method considers the correlation between both columns and rows. The mapping function is $C = Z^T AX$, and it can be seen as mapping to the rows first and then to the columns or to the columns first and then to the rows. At the same time the iteration ideology is proposed by G2D-PCA to obtain better results.

ND-PCA is proposed for modeling of high-dimension data. This method is based on HOSVD. At the same time, we treat the data as a high-dimension tensor. The method can solve the problem of high cost effectively, but it also has a large coefficient matrix as 2D-PCA method.

2.2. The Basic Knowledge of Tensor. Tensor can be treated as the expansion of matrix. Vector is a first-order tensor and matrix is a second-order tensor. So if we stack up several matrixes with the same dimension, we obtain the cubic array named third-order tensor. The analysis of high-order tensor uses the math operation as follows [6].

Suppose that X is an M-order tensor, $X \in R^{N_1 \times N_2 \times \cdots \times N_M}$, and N_i is the dimension of tensor X. The element of X is defined as $X_{n_1, n_2, \ldots, n_M}$, for $1 \le n_i \le N_i$, $1 \le i \le M$. The tensor product is defined as follows:

$$(X \otimes Y)_{n_1 \times n_2 \times \cdots \times n_M \times n'_1 \times n'_2 \times \cdots \times n'_M} = X_{n_1 \times n_2 \times \cdots \times n_M} Y_{n'_1 \times n'_2 \times \cdots \times n'_M}. \tag{1}$$

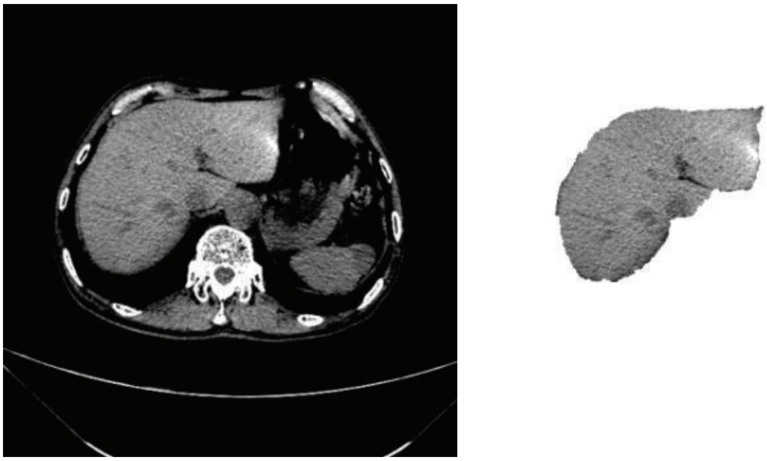
We can transfer the M-order tensor to a matrix by extending the N_dth vector of tensor X and put others after the N_d. The product function of tensor X and matrix U is shown as follows:

$$(X \times_d U)_{i_1 \times i_2 \times \cdots \times j \times i_{d-1} \times \cdots \times i_M} = \sum_{i_d} \left(X_{i_1 \times i_2 \times \cdots \times i_{d-1}} U_{j \times i_d} \right). \tag{2}$$

2.3. Construction of High-Order Tensor. In this paper, we construct high-order tensors based on fractal theory. Firstly, we use the method of box [7] and blanked [8] to calculate 4 groups of fractal feature, and then we establish high-order tensors based on the fractal feature and the texture feature pointing to each pixel.

2.3.1. The Calculation of Fractal Feature. We use the method of blanket and box to calculate the fractal feature of liver images which are segmented by the doctor. The liver image and its segmentation result are shown in Figure 3.

The first fractal feature is obtained by the blanket method. Firstly, we treat the images as a hilly terrain surface whose height from the normal ground is proportional to the gray level of the images. Then all points at distance ε from the surface on both sides create a blanket whose thickness is 2ε. The estimated surface area is the volume of blanket divided by 2ε. For different ε, the blanket area can be iteratively estimated as follows. The covering blanket is defined by its



(a) Original image

(b) Segmentation result

Figure 3: Liver segmentation.

upper surface u_ε and the lower surface d_ε, and we provide the gray level function $g(i,j)$, $u_0(i,j) = b_0(i,j) = g(i,j)$, for $\varepsilon = 1, 2, 3, \ldots$. Blanket surfaces are defined as follows:

$$u_\varepsilon(i,j) = \max\left\{u_{\varepsilon-1}(i,j)+1, \max_{|(m,n)-(i,j)|\le 1} u_{\varepsilon-1}(m,n)\right\},$$
$$b_\varepsilon(i,j) = \min\left\{b_{\varepsilon-1}(i,j)-1, \min_{|(m,n)-(i,j)|\le 1} b_{\varepsilon-1}(m,n)\right\}. \tag{3}$$

The volume of the blanket is defined as follows:

$$v_\varepsilon = \sum_{i,j}\left(u(i,j) - b_\varepsilon(i,j)\right). \tag{4}$$

The surface area can be defined as follows:

$$A(\varepsilon) = \frac{(v_\varepsilon - v_{\varepsilon-1})}{2},$$
$$A(\varepsilon) = F\varepsilon^{2-D}. \tag{5}$$

At last, the fractal feature D_1 can be described as (6), and v is the volume of the blanket:

$$D_1 = 2 - \log_\varepsilon^{(v_\varepsilon - v_{\varepsilon-1})/2F}. \tag{6}$$

The other fractal features are obtained by the method of box. It is to treat the gray level image $F(R \times R)$ as a box in 3-dimensional fractal curves. The image can be separated into several boxes $(S{\times}S)$, $\delta = S/R$. z is the gray level of the images; the plane surface XY can be separated into several grids. The maximum level and the minimum level of gray level of the image in the grid (i,j) can be treated as the kth and the 1st box, $n_\delta(i,j) = l-k+1$, and then we calculate the total number $N_\delta(F) = \sum n_\delta(i,j)$, and the fractal feature can be defined as (7). In this paper, we obtain D_2 to D_4 by giving different numbers of the boxes such as 4, 8, and 16:

$$D_B = \frac{\log N_\delta(F)}{\log(1/\delta)}. \tag{7}$$

There is a big texture difference in coarse level between different liver images, but the fractal dimension has a small change in smooth-faced images and a large change in shaggy images. So the fractal dimension is a useful feature for liver diseases classification.

2.3.2. The Construction of High-Order Tensor. In this paper, we use 50 groups of liver images of 512 × 512. After we extract four kinds of fractal features, we extract the texture features. We use all features we obtained to establish the high-order tensors.

2.4. The Construction of Statistical Fractal Dimension Based on GND-PCA. We provide a series of zero-mean value N-order tensor $A \in R^{I_1\times I_2\times\cdots\times I_N}$. And we need to gain a group of new N-order tensor $B \in R^{J_1\times J_2\times\cdots\times J_N}(J_n < I_n)$, and B needs to be closed to the original tensor as much as possible. Then we define tensor images by the texture and fractal features obtained from the segmented liver images. We use Tucker model [9] to reconstruct N-order tensor A by $U_{(n)}$, $U^{(n)} = J_n \times I_n$. The reconstruction of three-order tensor is shown in Figure 4.

The orthogonal matrix $U_{\text{opt}}^{(n)}$ can be obtained by minimizing the cost function C, which is shown as (8).

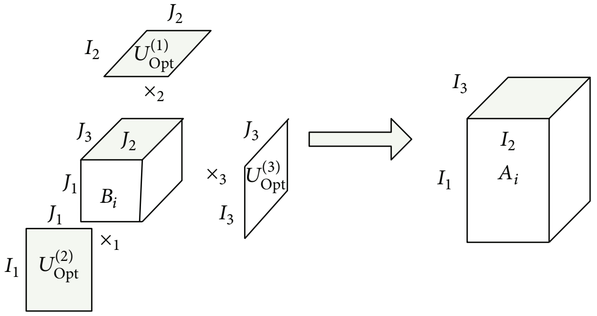


Figure 4: Reconstruction of third-order tensor image.

In $A_i \in R^{I_1 \times I_2 \times \cdots \times I_N}$, $i = 1, 2, \ldots, M$, M is the number of samples. A_i^* is the reconstructed tensor. There are two methods to minimize the cost function:

$$C = \sum_{i=1}^{M} \left\| A_i - A_i^* \right\|^2 = \sum_{i=1}^{M} \left\| A_i - B_i \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)} \right\|^2, \tag{8}$$

$$A_i^* = B_i \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)}. \tag{9}$$

The first is to minimize the cost function C directly, and we can calculate the orthogonal matrix by the function $B_i = A_i \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times \cdots \times_N U^{(N)^T}$. But it is difficult to calculate the function. The second method is to maximize C' shown as (10), and it is easier to calculate. In this paper, we used the second method:

$$C' = \sum_{i=1}^{M} \left\| A_i \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times \cdots \times_N U^{(N)^T} \right\|^2. \tag{10}$$

2.5. Construction of the Classification of Liver Diseases. In this paper, ACO is used to optimize SVM to train a liver diseases classifier. DAG structure for multiclassification is used to distinguish liver diseases.

2.5.1. Feature Selection Based on Liver Statistical Fractal Model. The samples consist of the core tensor of each tensor. We transfer the core tensor into a one-dimensional vector using the method of nonlinear data dimensionality reduction [10]. The training set is $D = \{d_1, d_2, \ldots, d_M\}$ (M is the total number of the samples), and the set of features is $T = \{t_1, t_2, \ldots, t_P\}$ (P is the total number of the features).

2.5.2. Feature Weighed. The number of gray level features we select is too much, and the number of fractal features is fewer than it. So we give a higher weight to the fractal features. A series of experiments showed that the classification accuracy is much better when the weight of fractal feature is 0.6 and the weight of gray level feature is 0.4.

2.5.3. Construction of Classifier Based on ACO-SVM. Some diseases such as cirrhosis and hepatic cyst are different from cancer. They are usually confused with cancer in CAD. SVM is always used in binary classification. If we want to classify 4 kinds of liver diseases using SVM, we should combine several SVMs. In this paper we use the method of directed acyclic graph (DAG-SVM ($k = 4$)) to realize the multiclassification of liver diseases, and DAG is shown in Figure 5.

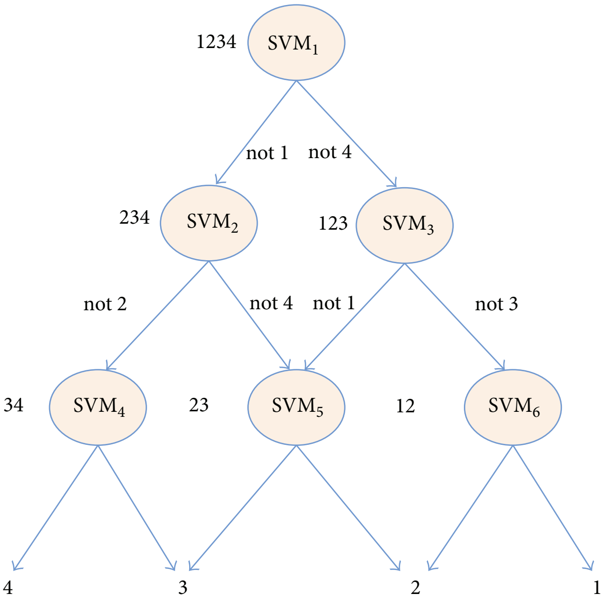


Figure 5: DAG-SVM, $k = 4$.

If we classify 4 kinds of liver diseases, we should use 6 SVMs. C is the penalty factor, and σ is the parameter of kernel function. In order to optimize these two parameters by ACO algorithm, C and σ must be discretized firstly. In this paper, the two parameters are discretized according to effective bits which are determined by experiences. The parameter C and σ has five effective bits, respectively. The value of each bit can be varied from 0 to 9. For C, its top digit is hundreds place, so its value ranges from 0 to 999.99. While for σ, its top digit is ones place, and thus its value ranges from 0 to 9.9999.

Then heuristic information $\eta(i, j)$ is set to 1. Classification accuracy is used to evaluate SVM performance, and therefore $\Delta\tau(i, j) = Q \cdot \text{Acc}$ is used in the global update process. Here Q is pheromone intensity and Acc is maximal classification accuracy in each cycle. The whole process is executed as follows.

Step 1. Discretizing parameters C and σ by the method as above.

Step 2. Initializing pheromone $\tau(i, j) = 1$ and pheromone increment $\Delta\tau(i, j) = 0$.

Step 3. Executing search process for the first best path.

(1) Laying ants at the origin of coordinates.

(2) Putting each ant to next city whose x coordinate is different from the previous visited cities randomly.

(3) Modifying pheromone of transfer path for each ant according to local update rule.

(4) Modifying pheromone of the path for the best ant according to global update rule if all the ants finish visiting 10 nodes, else returning to (2).

Step 4. Laying ants at coordinate origin again.

Step 5. Putting each ant to the next city chosen according to state transition rule.

Step 6. Modifying pheromone of transfer path for each ant according to local update rule. If ants finish tour, we jump to Step 7; otherwise we return to Step 5.

Step 7. Training a SVM classifier with C and σ obtained by each ant. We find out the best ant which produced the highest accuracy and modify pheromone for the best ant according to global update rule. If the accuracy meets termination condition or the times of loop are bigger than the maximum cycle times, we jump to Step 8; otherwise we return to Step 4.

Step 8. Outputting best C, σ and maximum accuracy.

3. Results and Discussion

We select 120 groups of liver images, 60 groups are normal liver, 20 groups are cirrhosis liver volume, 20 groups are cancer liver volume, and 20 groups are hydatoncus liver volume. There are 50 images in each group. The thickness of each image is 3 mm, and the resolution is 512 × 512. In 120 groups of images, we selected a half as training samples, the others as testing samples.

3.1. Reconstruction Results after GND-PCA. In this paper, we use leave-one-out method to test the generalization ability of models constructed without fractal features for liver volumes. One of all images is shown in Figure 6. The location of tumor is in the lower left corner of the liver image. Firstly, one volume is excluded from the training data which is used for the construction of the model, and then it is reconstructed by the training models for checking.

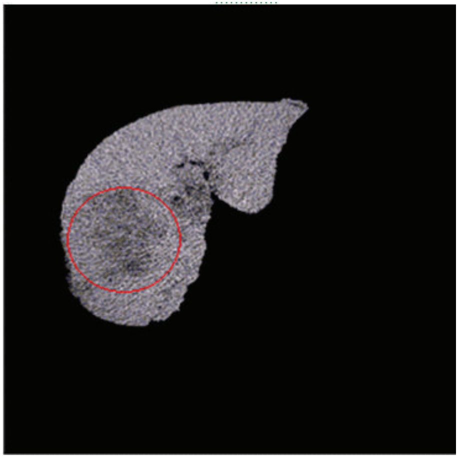


Figure 6: Liver cancer image.

The volume is reconstructed from $5 \times 5 \times 3$ to $300 \times 300 \times 30$ which is shown in Figure 7. In Figure 7, the first row is reconstruction of slice 3, the second row is reconstruction of slice 13, and the third row is reconstruction of slice 23. Column (a) is original liver image, column (b) is that the dimension of mode-subspace is $5 \times 5 \times 3$, column (c) is $100 \times 100 \times 10$, column (d) is $200 \times 200 \times 20$, column (e) is $300 \times 300 \times 30$, and column (f) is the reconstructed volume using eigenface by PCA as the contrastive method.

Since the dimension of the original volume is $512 \times 512 \times 50$, we can calculate the compressing rate for all cases. The compressing rate is 0.0006%, 0.7629%, 6.1035%, and 20.5994%. With the growth of the dimension of mode-subspace, reconstruction result is better. Because of overfitting, the method of PCA is worse than GND-PCA.

It needs less iteration times using GND-PCA which is shown in Figure 8, and the value of the cost function does not dramatically change after two iterations. Therefore, we set the iteration times of GND-PCA as two in our experiment.

In Figure 9, it shows the relationship between original volume and the reconstructed volume. Abscissa a is the mode-subspace of $5 \times 5 \times 3$, b is $100 \times 100 \times 10$, c is $200 \times 200 \times 20$, d is $300 \times 300 \times 30$, e is $400 \times 400 \times 40$, and f is $512 \times 512 \times 50$.

The normalized correlation grows with the growth of mode-subspace size. When the mode-subspace size is $512 \times 512 \times 50$, the normalized correlation is 1. It means that we can reconstruct the original volume without any errors. The normalized correlation can be defined as (11). $I(x, y, z)$ is the original tensor volume, and $I^*(x, y, z)$ is the volume after reconstruction:

$$NC = \frac{\sum_{x,y,z} I(x,y,z) I^*(x,y,z)}{\sqrt{\sum_{x,y,z} I^2(x,y,z)}\sqrt{\sum_{x,y,z} I^{*2}(x,y,z)}}. \quad (11)$$

3.2. Results of Classification. The result of each SVM in ACO-SVM multiclassifier is shown in Table 1. From the table we can see that the statistical fractal model has better accuracy than the statistical texture model without fractal feature.

Compared with other classifier, ACO-SVM with the weighed fractal feature has better accuracy which is shown in Figure 10. Classifier BPNN is BP neural network, and the accuracy is 69.23%. Classifier FL is Fisher linear classifier with 46.23% accuracy. Classifier KNN is k-Nearest Neighbor algorithm whose accuracy is 47.23%. Classifier SVM is the conventional SVM with 62.68% accuracy. Classifier ACO-SVM is the conventional ACO-SVM whose accuracy is 89.87%. Classifier F-ACO-SVM is the method which is ACO-SVM with fractal features, and the accuracy is 91.43%. Classifier WF-ACO-SVM is ACO-SVM with weighed fractal feature; the accuracy is 93.06%. As Figure 10 shows, ACO-SVM does better than others in classification. And when we use weighed fractal feature in our statistical fractal model, we can reach a better accuracy in liver diseases classification.

Table 1: Parameters optimization result of SVM for multiclassification using ACO.

Classifier	Best C	Best δ	NFD Acc	FD Acc	WFD Acc
ACO-SVM$_1$	622.57	1.3114	96.44%	97.64%	97.85%
ACO-SVM$_2$	783.96	5.2349	98.02%	98.52%	98.64%
ACO-SVM$_3$	100.230	1.2255	97.74%	99.87%	99.87%
ACO-SVM$_4$	14.020	0.2378	99.76%	99.83%	99.98%
ACO-SVM$_5$	984.69	1.1424	94.3%	96.57%	96.65%
ACO-SVM$_6$	876.78	1.0765	98.82%	99.33%	99.64%

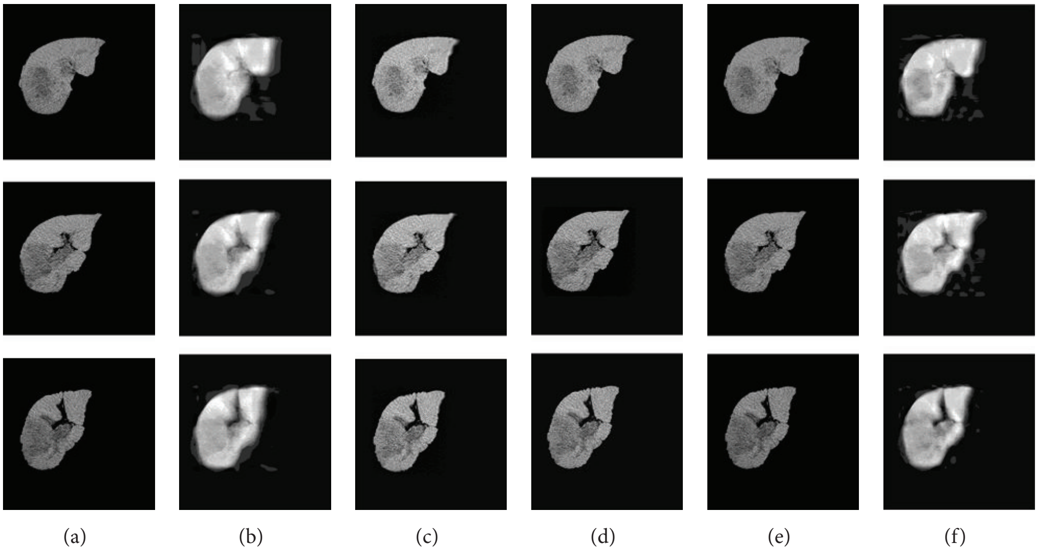


Figure 7: Reconstruction results.

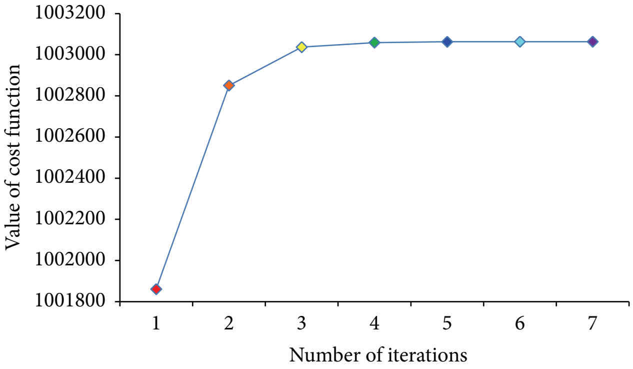


Figure 8: Convergence of GND-PCA.

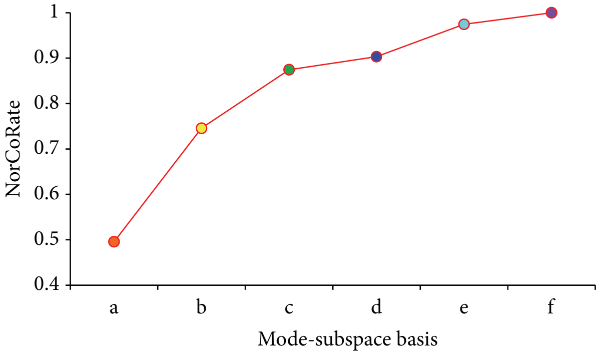


Figure 9: Normalized correlation between the original volume and the reconstructed volumes.

4. Conclusions

In this paper, we have presented the construction of high-order tensors with weighed fractal dimension feature and gray feature. And GND-PCA, which is a subspace learning method, has been used to get the core tensor from those high-order tensors and establish the statistical fractal model for the later classification. ACO-SVM has been used to train a liver image classifier. As an application for classifying liver diseases, the method using statistical fractal models based on GND-PCA and ACO-SVM achieved the better classification accuracy, because statistical fractal models based on GND-PCA can preserve the information of the original image as much as possible, and ACO can find the optimal parameters for SVM. In conclusion, under the condition of a small number of samples, the classifier of this paper can achieve the better recognition accuracy than others such as BPNN, the conventional SVM, and the conventional ACO-SVM.

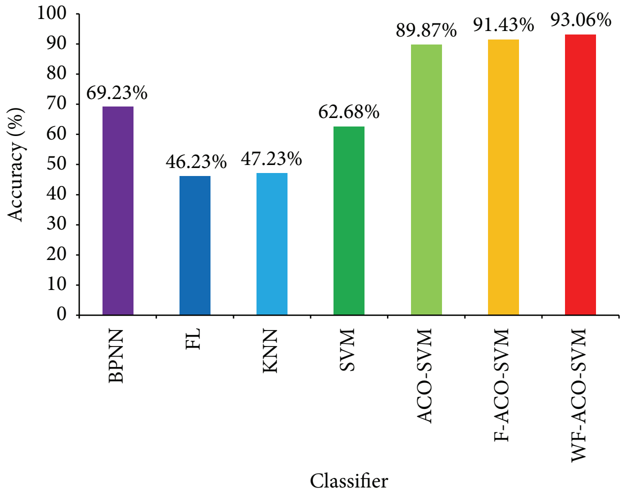


Figure 10: Result of multiclassification using seven classifiers.

Therefore the proposed method can improve the classification accuracy of liver diseases and assist doctors to diagnose liver diseases.

Acknowledgment

The research is supported by the National Natural Science Foundation of China (no. 61272176, no. 60973071).

References

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '91)*, pp. 586–591, June 1991.

[2] R. Xu and Y. W. Chen, "Generalized N-dimensional principal component analysis (GND-PCA) and its application on construction of statistical appearance models for medical volumes with fewer samples," *Neurocomputing*, vol. 72, no. 10–12, pp. 2276–2287, 2009.

[3] B. Liu, Z. F. Hao, and X. W. Yang, "Nesting support vector machinte for muti-classification [machinte read machine]," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, vol. 7, pp. 4220–4225, August 2005.

[4] J. C. Platt, N. Cristianini, and T. J. Shawe, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems*, vol. 12, no. 3, pp. 547–553, 2000.

[5] X. Liu, H. Jiang, and F. Tang, "Parameters optimization in SVM based-on ant colony optimization algorithm," *Advanced Materials Research*, vol. 121-122, pp. 470–475, 2010.

[6] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700–1715, 2007.

[7] S. Peleg, J. Naor, R. Hartley, and D. Avnir, "Multiple resolution texture analysis and classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 518–523, 1984.

[8] N. Sarkar and B. B. Chauduri, "An Efficient differential box-counting approach to compute fractal dimension of image," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 1, pp. 115–120, 1994.

[9] L. de Lathauwer, B. de Moor, and J. Vandewalle, "On the best rank-1 and rank-($R_1, R_2, \ldots, R_n$) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[10] H. Eghbalnia, A. Assadi, and J. Carew, "Nonlinear methods for clustering and reduction of dimensionality," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '99)*, vol. 2, pp. 1004–1009, July 1999.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 965318, 6 pages
http://dx.doi.org/10.1155/2013/965318

Research Article

The Quantitative Overhead Analysis for Effective Task Migration in Biosensor Networks

Sung-Min Jung,[1] Tae-Kyung Kim,[2] Jung-Ho Eom,[3] and Tai-Myoung Chung[1]

[1] *Department of Electrical and Computer Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Republic of Korea*
[2] *Department of Liberal Art, Seoul Theological University, Sosabon-dong, Sosa-gu, Bucheon-si, Gyeonggi-do 422-742, Republic of Korea*
[3] *Department of Military Studies, Daejeon University, 62 Daehakro, Dong-Gu, Daejeon-si 300-716, Republic of Korea*

Correspondence should be addressed to Tai-Myoung Chung; tmchung@ece.skku.ac.kr

Received 30 June 2013; Revised 22 August 2013; Accepted 22 August 2013

Academic Editor: Tai-hoon Kim

We present a quantitative overhead analysis for effective task migration in biosensor networks. A biosensor network is the key technology which can automatically provide accurate and specific parameters of a human in real time. Biosensor nodes are typically very small devices, so the use of computing resources is restricted. Due to the limitation of nodes, the biosensor network is vulnerable to an external attack against a system for exhausting system availability. Since biosensor nodes generally deal with sensitive and privacy data, their malfunction can bring unexpected damage to system. Therefore, we have to use a task migration process to avoid the malfunction of particular biosensor nodes. Also, it is essential to accurately analyze overhead to apply a proper migration process. In this paper, we calculated task processing time of nodes to analyze system overhead and compared the task processing time applied to a migration process and a general method. We focused on a cluster ratio and different processing time between biosensor nodes in our simulation environment. The results of performance evaluation show that task execution time is greatly influenced by a cluster ratio and different processing time of biosensor nodes. In the results, the proposed algorithm reduces total task execution time in a migration process.

1. Introduction

A biosensor network is generally composed of many biosensor nodes and one base station. Biosensor nodes are distributed on a human body or wearable devices, and the base station is located at outside of biosensor networks. Biosensor nodes monitor various biological parameters such as body temperature, blood pressure, and blood glucose level. They transmit these gathered data to the base station, and the base station derives meaningful results from the processed data. Finally, the base station sends these results to user's device or to a hospital through the internet as shown in Figure 1 [1].

In general, main components of a biosensor node are a sensing unit, a processing unit, a transceiver, and a power unit [2, 3]. Biosensor nodes can monitor specific parameters by using sensing units, and gathered information is delivered to the processing unit. The processing unit is composed of a processor, storage, and memory. These subunits manage a procedure to analyze collected information and relay it to other biosensor nodes. The collected information by biosensor nodes is typically medical data, so it is sensitive and privacy information. This information should not be exposed to a malicious user, and biosensor nodes should process it in real time [4, 5]. Because of their small size, biosensor nodes have little computational power, limited capacity of memory, and restricted battery. Thus, an attack can easily decrease availability of biosensor nodes, so it makes it impossible that they relay the important information to user in real time [6, 7]. In this case, a suitable migration process has to be used to solve that problem. The migration means the process of transferring tasks from nodes with heavy overhead to other nodes with enough capabilities.

In this paper, we propose a useful algorithm to quantitatively analyze system overhead. Simulation results show that network performance is greatly influenced by a cluster ratio and different performance of biosensor nodes. Also,

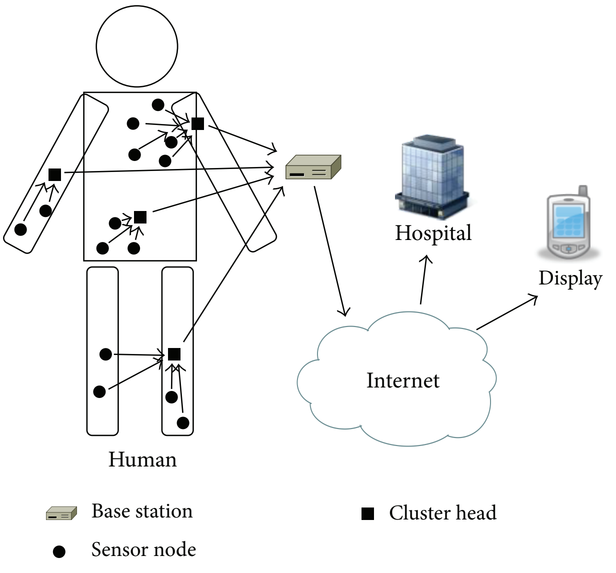


Figure 1: The concept of biosensor networks.

using the proposed algorithms, the total task execution time is reduced compared with a general process. The remainder of this paper is organized as follows. In Section 2, we discuss restricted resources of biosensor nodes and the reason to use a clustering scheme in our system model. In Section 3, we present mathematical analysis to calculate total task execution time and system overhead. In Section 4, we evaluate the proposed algorithm with several parameters. Finally, Section 5 concludes this paper.

2. The Clustering Scheme in System Model

Biosensor nodes are generally very small, so the use of computing resources is limited. In particular, it is impossible to replace or recharge the power unit, so it is important to reduce energy consumption in the biosensor network. Since it is necessary to make uniform energy consumption to all biosensor nodes, we need to use a hierarchical routing protocol [8, 9]. This protocol uses a cluster which indicates a logical group of biosensor nodes, and the cluster is managed by the leader node called a cluster head. Before biosensor nodes gather data, cluster heads are selected, and clusters are formed around these cluster heads in the hierarchical routing protocol. The cluster heads are responsible for gathering information from all biosensor nodes in their cluster. After gathering information, cluster heads perform data aggregation to reduce data size and transmit results to the base station. The role of the cluster head is periodically rotated to prevent energy depletion of particular biosensor nodes. Therefore, our system model uses the hierarchical routing protocol to reduce energy consumption. Figure 2 shows the clustering scheme.

Biosensor nodes are generally distributed in wearable equipment or on a human body. It is assumed that the base station knows the network topology. Also, the base station has sufficient battery and processing capability. There are two kinds of sensors such as cluster heads and normal biosensor nodes. Biosensor nodes have formed the cluster by using the cluster head selection algorithm [10, 11]. When overhead is occurs in some nodes, a suitable migration process could be used to reduce overhead. In other words, tasks are moved from biosensor nodes with large overhead to the other nodes with sufficient resources. The task execution time can represent system overhead, so we calculate and compare it to analyze system overhead in our system model.

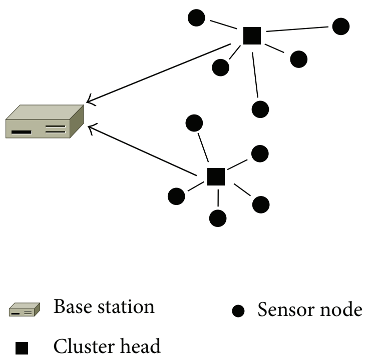


Figure 2: The clustering scheme.

First, we set that task execution time is the sum of processing time and communication time as shown in (1). Processing time indicates the time required to process the tasks in each biosensor node. Thus, total processing time of all tasks depends on the number of active biosensor nodes. Communication time means the time required to transmit from each biosensor nodes to the base station. There are two types of communication time in the hierarchical routing protocol. One is the transmission time from biosensor nodes to cluster heads, and the other is transmission time from cluster heads to the base station:

$$\begin{aligned}\text{Task execution time} = {} & \text{Processing Time} \\ & + \text{Communication Time.}\end{aligned} \tag{1}$$

The biosensor nodes check processing time in regular period and record the fastest and slowest value. Let T_f denote the fastest processing time and let T_s denote the slowest processing time which is required to process a unit task. It is assumed that the processing time follows uniform distribution from T_f to T_s. Figure 3 shows its probability density function. Let N_i denote the initial number of biosensor nodes to process all the tasks in this function.

In this function, the expected value is as shown in (2) by a uniform distribution rule. If there is no consideration of a migration process to calculate the task execution time, then (2) is used to calculate the task execution time of biosensor nodes:

$$E[X] = \frac{T_f + T_s}{2}. \tag{2}$$

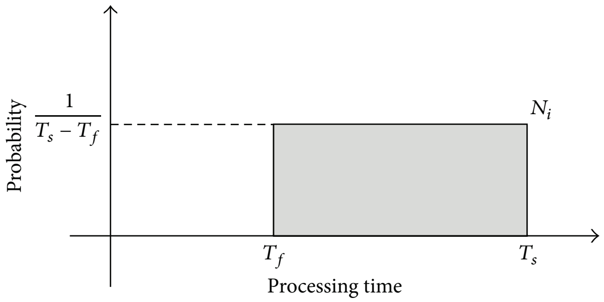


Figure 3: The probability density function.

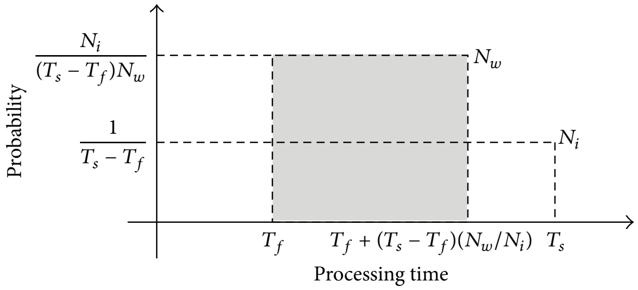


Figure 4: The new probability density function.

3. The Proposed Algorithm to Analyze System Overhead

We focus on the number of biosensor nodes and assume that the system performance is linearly improved by the number of nodes [12]. Thus, the number of active biosensor nodes is different by system overhead. When some biosensor nodes have heavy overhead, we can solve this problem to use a migration process. Let N_i denote the initial number of biosensor nodes. After we move tasks from N_w/N_i of nodes to the other nodes, the number of active biosensor nodes becomes N_w.

As a result, the processing time becomes a new uniform distribution as shown in Figure 4. It is distributed from T_f to $T_f + (T_s - T_f)(N_w/N_i)$. Equation (3) shows the expected value of its probability density function:

$$E[X] = T_f + \frac{(T_s - T_f)}{2N_i} \cdot N_w. \tag{3}$$

We should consider the number of tasks to calculate system overhead. Let N_e denote the number of tasks. Since the number of active biosensor node is N_w, each biosensor node has to process N_e/N_w tasks. Equation (4) indicates the expected value of processing time in each biosensor node:

$$T_{\text{process}} = \frac{N_e}{N_w}\left(T_f + \frac{(T_s - T_f)}{2N_i} \cdot N_w\right). \tag{4}$$

As mentioned in the previous section, our system model uses a clustering scheme, and the number of biosensor nodes is similar in each cluster. Let R_c denote cluster ratio; then, the number of clusters will be $N_w \times R_c$, and the number of biosensor nodes in one cluster will be $1/R_c$.

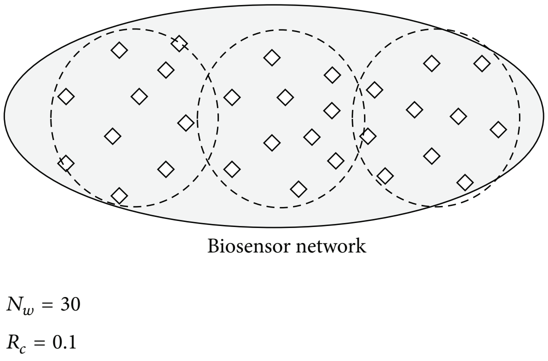


Figure 5: The number of cluster by a cluster ratio.

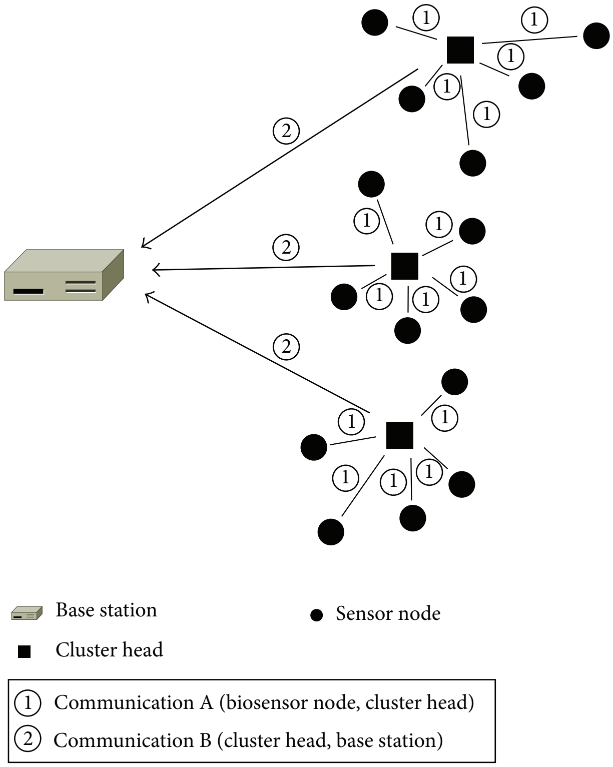

1
2
Base station
Sensor node
Cluster head
1 Communication A (biosensor node, cluster head)
2 Communication B (cluster head, base station)

Figure 6: The two types of communication.

For example, the number of biosensor nodes is 30, and the cluster ratio is 0.1; then, the number of clusters is 3 (30 × 0.1), and the number of biosensor nodes is 10 (1/0.1) in each cluster as shown in Figure 5.

We calculate the communication time in the biosensor network. Let T_t denote transmission time of unit packet. There are two types of communication as shown in Figure 6. First, the communication time from a biosensor node to a cluster head in each cluster is represented as the product of the data transmission time and the number of biosensor nodes in each cluster. Biosensor nodes sequentially transmit results according to the order, and the communication time from biosensor nodes to a cluster head can be presented as

Table 1: Parameters for analysis.

Parameters	Values	Descriptions
N_e	100	The number of tasks
N_w	1 ~ 30	The number of active biosensor nodes
R_c	0.1, 0.2, 0.3	The cluster ratio
T_f	0.001	The fastest processing time of a biosensor node
T_s	Variable	The slowest processing time of a biosensor node
N_i	30	The initial number of biosensor nodes
T_t	0.0041 sec	The transmission time

$T_t \times (1/R_c)$. Second, the communication time from a cluster head to the base station is expressed as the product of data transmission time and the number of cluster heads. Because cluster heads also sequentially send results to the base station, communication time in second case is expressed as $T_t \times N_w \times R_c$. Equation (5) indicates the total communication time:

$$T_{\text{communication}} = T_t \cdot \frac{1}{R_c} + T_t \cdot N_w \cdot R_c. \tag{5}$$

Finally, the sum of (4) and (5) represented the time required to process all tasks and transmit to the base station when the number of active biosensor nodes is changed from N_i to N_w:

$$T_{\text{all}} = \frac{N_e}{N_w}\left(T_f + \frac{\left(T_s - T_f\right)}{2N_i} \cdot N_w\right) + T_t \cdot \frac{1}{R_c} + T_t \cdot N_w \cdot R_c. \tag{6}$$

4. Performance Evaluation

We evaluate the total task execution time by (6) in a biosensor network. We use the parameter values listed in Table 1 for our analysis of task execution time. We set the number of tasks (N_e) to 100 and the initial number of biosensor nodes (N_i) to 30. The number of active biosensor nodes (N_w) is from 1 to 30. The fastest processing time of a biosensor node (T_f) is 0.001 seconds. We set the unit message length to 128 byte and transmission speed to 250 kbps. Therefore, it takes about 0.0041 seconds to process the unit message length so we set the transmission time (T_t) to 0.0041 seconds. Since the communication range of a biosensor network is very small and there is very little impact on the performance by distance between biosensor nodes, the distance is ignored in our performance evaluation.

Based on these simulation parameters, we evaluate total task execution time according to change of a cluster ratio and the slowest processing time of a biosensor node in a migration process.

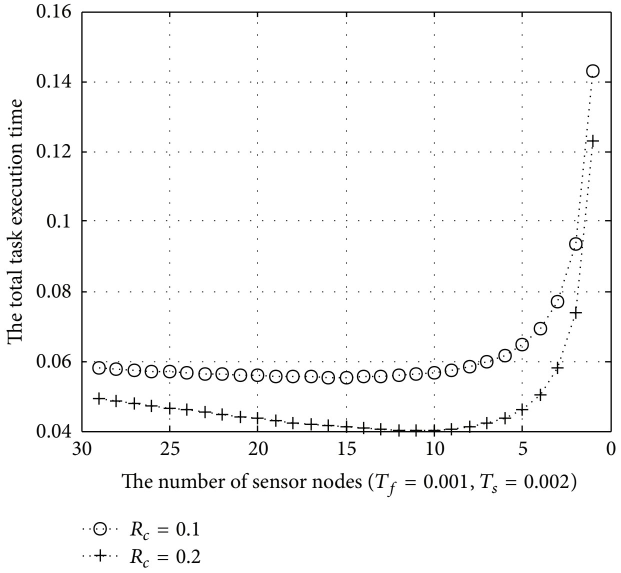


Figure 7: The total task execution time (T_s = 0.002).

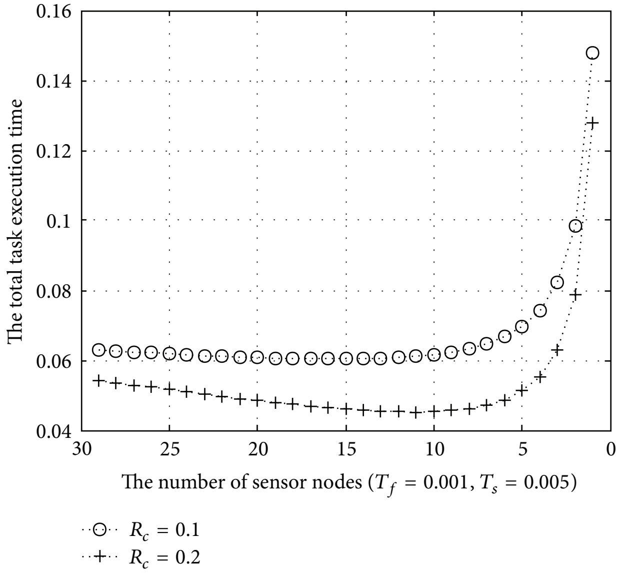


Figure 8: The total task execution time (T_s = 0.005).

In Figures 7 and 8, we set T_s to 0.002 and 0.005 seconds, respectively. Also, we calculate task execution time according to the change of a cluster ratio (R_c). R_c is 0.1 and 0.2 in each figure. Figure 7 shows the result between the total task execution time and N_w. In this evaluation, T_f is set as 0.001, and T_s is set as 0.002 seconds. At first, the total task execution time decreases as the number of active biosensor nodes decreases. However, the task execution time increases after a certain number of biosensor nodes. It is 15 in case R_c is 0.1 and 10 in case R_c is 0.2 in Figure 7.

Figure 8 shows the result when T_f is 0.001 and T_s is 0.005. We can recognize that total task execution time is influenced by T_s as compared with the result of Figure 7. Overall, the task execution time also increases as the different processing time increases. After the number of biosensor nodes becomes about 30% of the initial number of them, the task execution time increases rapidly. Thus, we can know that system overhead is tolerable until this point. As the cluster ratio increases, the change of the total task execution time

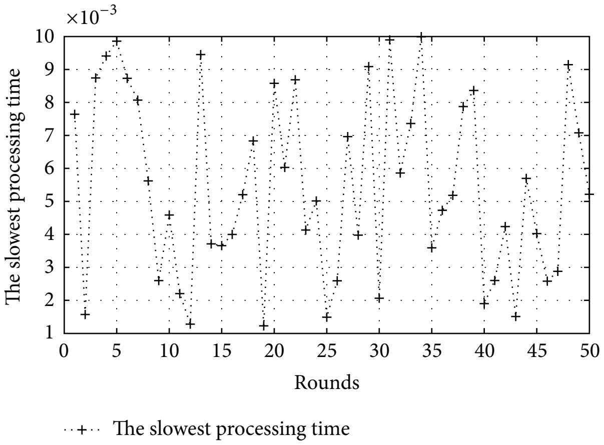


Figure 9: The slowest processing time ($T_f = 0.001$).

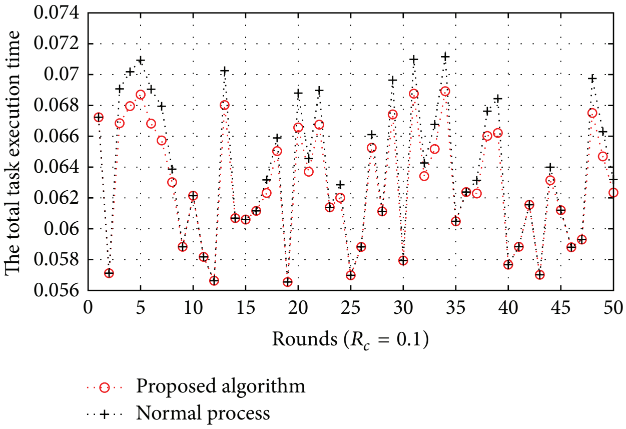


Figure 10: The total task execution time ($R_c = 0.1$).

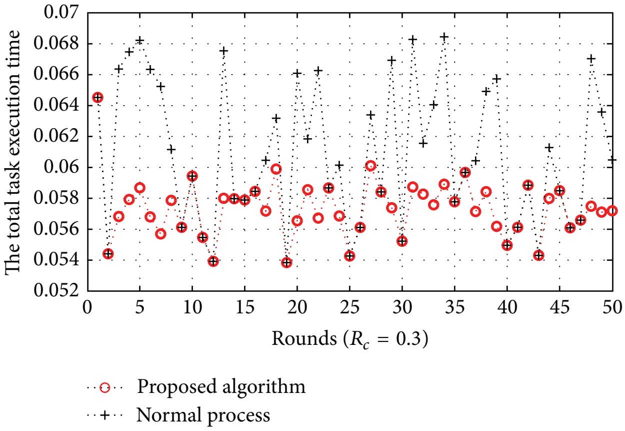


Figure 11: The total task execution time ($R_c = 0.3$).

decreases. As shown in the graph in Figures 7 and 8, total task execution time is affected by a cluster ratio and difference of processing time among biosensor nodes. Thus, we have to control these parameters to manage biosensor networks efficiently.

When biosensor nodes have heavy overhead, we can solve this problem by moving tasks from these nodes to other nodes with enough resources. The number of active biosensor nodes is changed, and it is needed to accurately calculate overhead in biosensor network. Also, we evaluate the total task execution time by (6) as the change of the slowest processing time of biosensor nodes.

Figure 9 shows the change of the slowest processing time in our simulation. There are different values from 0.001 to 0.010 seconds. We set T_f to 0.001 seconds. At each round we compared the task execution time applied to migration scheme and the task execution time applied to general method.

Figure 10 shows that the total execution time as T_s is changed when R_c is 0.1. The number of active nodes is 30. If T_s is greater than or equal to nine times of T_f, then tasks in 30% of all biosensor nodes move to the other biosensor nodes in our proposed algorithm. In the same way, T_s is greater than or equal to seven times and five times of T_f, and we move the tasks in 20% and 10% of all biosensor nodes, respectively. In Figure 10, the sum of execution time applied to normal process at each round is 3.1795 seconds, and the sum of execution time applied to a migration process is 3.1372 seconds. We can reduce the total task execution time by using our proposed algorithm, and the system performance has been improved by 1.35% in this case.

Figure 11 shows the result between the total execution time and T_s at each round when R_c is 0.3. If we do not consider a migration process, the total execution time is 3.0440 seconds. Conversely, if we use our proposed algorithm, the total execution time is 2.8677 seconds. Also, we can decrease the total execution time, and the system performance has been improved by 6.15% in this case.

When some biosensor nodes have large overhead, proper migration process is needed to manage the biosensor network efficiently. We suggest the algorithm to quantitatively analyze the total task execution time for effective task migration. The proposed algorithm is useful to apply a proper migration process, and the simulation result shows that it efficiently reduces the total task execution time.

5. Conclusion

A biosensor network is composed of many biosensor nodes with sensing, computation, and wireless communication capabilities to collect biological parameters of a human body. Biosensor nodes collect these parameters and relay them to other biosensor nodes or to the base station. Biosensor nodes have restricted resources due to their small size. Thus, the biosensor network is vulnerable to an external attack. When the malicious user attacks the system, some nodes have heavy overhead and the overall system performance will be degraded. We can solve this problem to apply a proper migration process.

In this paper, we propose the quantitative solution to figure out task execution time. Also, we compare the total task execution time applied to a migration process and a general method. The results of performance evaluation show that total execution time is affected by a cluster ratio and processing time between biosensor nodes. Therefore, it is needed to manage a cluster ratio and difference of processing time against an attack. Our proposed algorithm reduces the total task execution time by using a proper migration process. In this scheme, the method to calculate the processing time of biosensor nodes is not considered. Therefore, we are going to research to accurately calculate the processing time for more accurate simulation.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210).

References

[1] Y. Zhu, S. L. Keoh, M. Sloman, and E. C. Lupu, "A lightweight policy system for body sensor networks," *IEEE Transactions on Network and Service Management*, vol. 6, no. 3, pp. 137–148, 2009.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[3] Z. Hussain, M. P. Singh, and R. K. Singh, "Analysis of lifetime of wireless sensor network," *International Journal of Advanced Science and Technology*, vol. 53, pp. 117–126, 2013.

[4] S. K. Dhurandher, S. Misra, A. Dhawan, and A. Tiwari, "Efficient solutions to various routing issues involved in mobile ad hoc bio-sensor networks: applying appropriate motion trajectories," *IET Communications*, vol. 3, no. 5, pp. 830–845, 2009.

[5] S. Saleem, S. Ullah, and K. S. Kwak, "Towards security issues and solutions in Wireless Body Area Networks," in *Proceedings of the 6th International Conference on Networked Computing (INC '10)*, pp. 349–352, May 2010.

[6] J. Xu, J. Wang, S. Xie, W. Chen, and J.-U. Kim, "Study on intrusion detection policy for wireless sensor networks," in *Proceedings of the International Journal of Security and Its Applications (IJSIA '13)*, vol. 7, pp. 1–6, 2013.

[7] V. B. Balasubramanyn, G. Thamilarasu, and R. Sridhar, "Security solution for data integrity in wireless BioSensor networks," in *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07)*, pp. 79–82, June 2007.

[8] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–27, 2004.

[9] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.

[10] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[11] R. Sheikhpour and S. Jabbehdari, "A two-level cluster based routing protocol for wireless sensor networks," in *International Journal of Security and Its Applications (IJSIA '12)*, vol. 45, pp. 19–30, 2012.

[12] S. Yeo and H.-H. S. Lee, "Using mathematical modeling in provisioning a heterogeneous cloud computing environment," *Computer*, vol. 44, no. 8, Article ID 5740825, pp. 55–62, 2011.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 274193, 16 pages
http://dx.doi.org/10.1155/2013/274193

Research Article

Evaluation of Stream Mining Classifiers for Real-Time Clinical Decision Support System: A Case Study of Blood Glucose Prediction in Diabetes Therapy

Simon Fong,[1] Yang Zhang,[1] Jinan Fiaidhi,[2] Osama Mohammed,[2] and Sabah Mohammed[2]

[1] *Department of Computer and Information Science, University of Macau, Macau, China*
[2] *Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada P7B 5E1*

Correspondence should be addressed to Simon Fong; ccfong@umac.mo

Received 27 June 2013; Accepted 3 August 2013

Academic Editor: Tai-hoon Kim

Earlier on, a conceptual design on the real-time clinical decision support system (rt-CDSS) with data stream mining was proposed and published. The new system is introduced that can analyze medical data streams and can make real-time prediction. This system is based on a stream mining algorithm called VFDT. The VFDT is extended with the capability of using pointers to allow the decision tree to remember the mapping relationship between leaf nodes and the history records. In this paper, which is a sequel to the rt-CDSS design, several popular machine learning algorithms are investigated for their suitability to be a candidate in the implementation of classifier at the rt-CDSS. A classifier essentially needs to accurately map the events inputted to the system into one of the several predefined classes of assessments, such that the rt-CDSS can follow up with the prescribed remedies being recommended to the clinicians. For a real-time system like rt-CDSS, the major technological challenges lie in the capability of the classifier to process, analyze and classify the dynamic input data, quickly and upmost reliably. An experimental comparison is conducted. This paper contributes to the insight of choosing and embedding a stream mining classifier into rt-CDSS with a case study of diabetes therapy.

1. Introduction

Clinical decision support system (CDSS) is a computer tool which broadly covers autonomous or semiautonomous tasks ranging amang symptoms diagnosis, analysis, classification, and computer-aided reasoning on choosing some appropriate medical care or treatment. Quoting from [1], a CDSS can be defined as "a system that is designed to be a direct aid to clinical decision-making in which the characteristics of an individual patient are matched to a computerized clinical knowledge base, and patient-specific assessments or recommendations are then presented to the clinician(s) and/or the patient for a decision." As concise as this description goes, the brain of a CDSS is an automatic classifier which usually is a mathematically induced logic model. The model should be capable of mapping the relations between input events (usually are medical symptoms) and some predefined verdicts in the forms of medical advices/treatments. In other words, the classifier is delegated to predict or infer what the medical consequence will be, given the emerging events (sometimes medical interventions or prescriptions) as well as historic data that have been collected over time and induced into a classification model. The suggested medical consequences or so-called assessments and advices by the CDSS would be objectively recommended to a doctor for subsequent actions.

The underlying logics associated at the classifier of a CDSS are captures of knowledge or understanding between some attribute variables and the conclusion classes. The logics are represented either as some nonlinear mappings like numeric weights in an artificial neural network (black-box approach) or in some predicate-logic like IF-THEN-ELSE rules [2] known as clinical pathways. Traditionally the underlying logics are derived from a population of historic medical records, hence the induced model is generalized, versus which an individual new record can be tested for decision. The historic data are accumulated over time into a sizable volume for training the classification model. The records usually are digitized in electronic format and organized in a database [3]. Every time when a new instance of record is added, the classifier however needs to be rebuilt, in order to refresh its underlying logics to include the recognition of the new record. This learning approach is called "batch-mode" which inherits from the old design of many machine learning algorithms like greedy-search or partition-based decision tree: a model is trained by loading in the full set of data, and the decision tree is built by iteratively partitioning the whole data into hierarchical levels via some induction criteria. The short-comings of batch-mode learning have been studied and reported in [4], specifically its time latency in rebuilding the classification model whenever an additional record arrives.

The batch-mode learning kind of classifiers may work well with most of the CDSS when the updates over the ever-increasing volume of the medical records can be set periodic, and no urgency of a CDSS output is assumed. For example, the update for the CDSS classifier can happen at midnight when the workload of the computing environment is relatively low, and allowing for delay in inclusion of the latest records over 24 hours is acceptable for its use prior to the update. Most of the CDSS designs function according to this batch-mode approach (more details in Section 2) for nonemergency and perhaps nontime-critical decision-support applications, such as consultation by a general practitioner, nutrient advisor, and nursing care [5]. In general, CDSSs that adopt the batch-model learning while adequately meet the usage demands are those characterized by data that do not contain many fast-paced episodes and usually do not carry severe impacts. So there is little difference in its efficacy regardless the very latest records which are included in the training of the classifier or not. Examples are those decision applications over the data that evolve relatively slowly, which include but are not limited to common diseases that largely affect the world's population, cancers of which their treatments and damages may take months to years along the clinical timespan to take effect. In these cases, traditional CDSS with batch mode learning suffice their roles.

In contrast, a new type of CDSS called real-time clinical decision support system (rt-CDSS), as its name suggests, is able to analyze fast-changing medical data streams and can predict in real-time based on the very latest input events. Examples of fast-changing medical data are live feeds of vital biosignals from monitoring machines, like EEG, ECG, and EMG, as well as respiratory rate and blood oxygen level which are prone to change drastically in minutes or seconds. rt-CDSS usually is dealing with critical medical conditions, such as ICU, surgery, A&E, or mobile onsite rescue, where a medical practitioner opts for immediate decision-support by the rt-CDSS instrument based only on the latest measurements of his vital conditions. The information of vital conditions of the patient evolves very quickly during the course of operation, and it does matter of course in life and death.

As forementioned, a classifier is central to the design of CDSS, and the traditional batch-mode learning method obviously runs short for supporting a real-time CDSS due to its model refresh latency. As it was already pointed out in [6] the latency would increase probably exponentially as the training data size grows to certain amount; it means the classifier will become increasingly slow as fresh data continue to stream in, because of the continually training. In order to tackle with the drawback of batch-mode learning, a new breed of data mining algorithms called data stream mining has been recently invented [7] whose algorithms are founded on incremental learning. In a nutshell, incremental learning is able to process potentially infinite amount of data very quickly; the model update is incremental such that the underlying logics are refreshed reactively on the fly upon new instances, without the need of scanning through the whole dataset that embraces the new data repeatedly.

In the advent of incremental learning, new classifiers started to bring impacts into the biomedical research community. Some unprecedented real-time CDSS designs are made possible, in commercial prototype [8, 9] and in academic research [10–12]; even the developments are still in progress. These designs are characterized by having a real-time reasoning engine that is able to respond with fast and accuracy to clinical recommendation. The real-time decision generated by rt-CDSS is actually interpreted as a computer-inferred prediction from the given current condition of the patient that leads to further reasoning with an aid of a knowledge base, rather than a final decision confirmed by some authoritative human user. Generally there are two phases in the design of rt-CDSS, as shown in Figure 1.

Live data feeds deliver real-time events to the classifier which learns the new data incrementally and be able to map the current situation to one of the predefined class labels as predicted outcomes. The predicted outcomes by the classifier are subsequently passed the reasoning engine that connects to a knowledge base for generating medical advices in real time, usually event driven. The reasoning engine could be implemented in various ways such as case-based reasoning or a novel approach [10] that embedded pointers at the decision tree leaves of the classifier, leading to some predefined guidelines of medical cure.

The focus of this paper however is on the real-time classifier, while the reasoning part of the rt-CDSS has already been discussed in [10]. The prediction by the real-time classifier here in the medical context is defined as a quantitatively guessed outcome that is likely to happen in the near future given the information of the current condition and the recent condition of the patients as well as the drug intake or clinical intervention, if any. Based on the predicted outcome, the rt-CDSS fetches the best option of cure correspondingly from a given knowledge base.

In our previous paper, we proposed a framework of rt-CDSS [10]; Very Fast Decision Tree (VFDT) was adopted as a candidate of a real-time classifier in the system design, because VFDT is classical and the most original type of stream-based classifiers [11]. Successively there are other variants modified from VFDT. Although VFDT is believed to be able to fulfill the role of real-time classifier in rt-CDSS, at least theoretically and conceptually, the performance has not been validated yet. As real-time classifier is the core of rt-CDSS, its performance must be able to fulfill the stringent criteria such as very short latency, very high accuracy, and very high consistency/reliability. This paper contributes to the insight of selecting and embedding a stream mining classifier

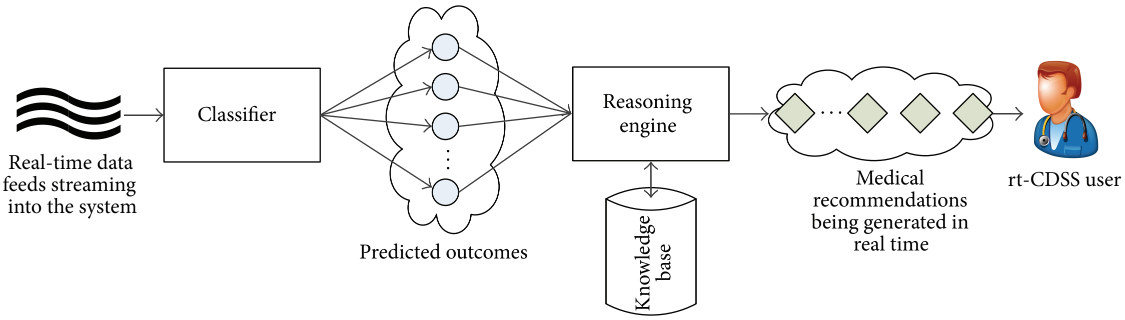

Real-time data feeds streaming into the system
Classifier
Predicted outcomes
Reasoning engine
Knowledge base
Medical recommendations being generated in real time
rt-CDSS user

Figure 1: Block diagram of a general rt-CDSS system.

into rt-CDSS with a case study of diabetes therapy that represents a typical real-time decision-making application scenario.

As a case study for comparative evaluation of classifiers for rt-CDSS, a computer-aided therapy for insulin-dependent diabetes mellitus patients is chosen to simulate a real-time decision making process in a scenario of dynamic events. The blood glucose level of diabetes patients often needs to be closely monitored, and it remains as an open question on how much the right dosage of insulin and the frequency of the doses should be given to maintain an appropriate level of blood glucose. This depends on many variables including the patient's body, lifestyle, food intake, and, of course, the variety of insulin doses. Along with this causal relationship between the predicted blood glucose levels and many contributing factors, multiple episodes can happen that may lead to different outcomes at any time. This is pertinent for testing the responsiveness and accuracy of the stream classifier considering that the episodes are the input values which may spontaneously evolve over time; the prediction is the guess work of the outcome based on the recent episodes.

The objective of this paper is twofold. We want to find out the most suitable classifier for rt-CDSS, and therefore we compared them in a diabetes therapy scenario. Also we want to test the performance of the classifier candidate all-rounded with a real-time case study, as a preliminary step to validate the efficacy of the rt-CDSS as a whole. Hence the study reported in this paper could serve as a future pathway for real-time CDSS implementation. The rest of the paper is structured as follow. An overview of classifiers that are used in CDSS is introduced in Section 2. The experiment to be conducted is described in details in Section 3. The second phase of rt-CDSS namely the decision inference is given in Section 4. Section 5 concludes the paper.

2. Related Work

In the literature there are quite a number of clinical decision support systems being proposed for different uses. It is cautious that the type of the classifier has a direct effect on the real-time ability of CDSS. In this section, some related work on different medical applications is reviewed with the aim of pointing out the shortcomings of some legacy research approaches pertaining to rt-CDSS.

A recent report [13] discusses the potential of CDSS technology in breast cancer excerpted from multidisciplinary team meetings, as a synergy, by the National Health Service (NHS), in the United Kingdom. The report essentially highlighted the importance of CDSS in structural and administrative aspects of cancer MDTs such as preparation, data collection, presentation, and consistent documentation of decisions. But at an advanced level, the services of a CDSS should exceed beyond the use of clinical databases and electronic patients' records (EPRs), by actively supporting patient-centred, evidence-based decision-making. In particular, a beta CDSS called multidisciplinary team assistant and treatments elector MATE, is being developed and trialed at the London Royal Free hospital. MATE is equipped with functionalities of prognostication tools, decision panel where system recommendations and eligible clinical trials are highlighted in colors, and the evidential justification for each recommended option.

In the report, it was stated like a wish list that an advanced CDSS is able to evaluate all available patient data in real time, including comorbidities, and offer prompts, reminders, and suggestions for management in a transparent way. The purpose of the report is to motivate further research along the direction of advanced CDSS. Although it is unclear about which classifier that is built into MATE, incremental type of classifier would well be useful if it were to receive and analyze real-time data streams with very quick responsiveness.

On the other hand, a classical algorithm, namely, artificial neural network (ANN) has been widely used in CDSS. ANNs apply complex nonlinear functions to pattern recognition problems and generally yield good results. Szkoła et al. [14] built CDSS for laryngopathies by extending ANN algorithms that are based on the speech signal analysis to recurrent neural networks (RNNs). RNNs can be used for pattern recognition in time series data due to their ability of memorizing some information from the past. The data that the system deals with are speech signals of patients. Speeches are usually spoken intermittently, and they are hardly continuous data streams. In their case, rt-CDSS might not be applicable. The other group, led by Walsh et al, proposed an ensemble of neural networks for building a CDSS [15] for bronchiolitis for infants and toddlers. They showed that using an ensemble that works like a selection committee usually outperforms single neural networks.

There is another common type of conventional classification algorithms based on decision rules, for deciding how an unseen new instance is to be mapped to a class. Gerald et al. developed a logistic regression model showing those variables that are most likely to predict a positive tuberculin skin test in contacts of tuberculosis cases. Their paper [16] shows that a decision tree is developed into a CDSS for assisting public health workers in determining which contacts are most likely to have a positive tuberculin skin test. The decision tree model is built by aggregating 292 consecutive cases and their 2,941 contacts seen by the Alabama Department of Public Health over a period of 10 months in 1998.

Another similar decision-support system called MYCIN [17] embeds decision rules into an expert system that provides interactive consultation. The decision rules are built into a simple inference engine, with a knowledge base of approximately 600 rules. MYCIN provided a list of possible culprit bacteria ranked from high to low based on the probability of each diagnosis, its confidence in each diagnosis' probability, the reasoning behind, and its recommended course of drug treatment. In spite of MYCIN's success, there is a debate about its classifier which essentially is an ad hoc sparked off. The rules in MYCIN are established on an uncertainty framework called "certainty factors." However, some users are skeptical about its performance for it could be affected by perturbations in the uncertainty metrics associated with individual rules, suggesting that the power in the system was coupled more to its knowledge representation and reasoning scheme than to the details of its numerical uncertainty model [18]. Classical Bayesian statistics should have been used as suggested by some doubters.

Iliad [19] which is a medical expert system software implementing Bayesian network as classifier has been developed by the University of Utah, School of Medicines, Department of Medical Informatics. In Iliad the posterior probabilities of various diagnoses are calculated by Bayesian reasoning. It was designed mainly for diagnosis in internal medicine. Currently it was used mainly as a classroom teaching tool for medicate students. Its power especially the Bayesian network classifier has not been leveraged for stream-based rt-CDSS.

Of all the well-known CDSS reviewed so far above, there is no suggestion indicating that they are operating on real-time live data feed; the data that they work on are largely EPRs, both patient-specific and of propensity, and perhaps coupled with clinical laboratory tests. Nevertheless, architectures of rt-CDSS namely, BioStream, [20], Aurora [21], and other monitoring devices [22] have been proposed which are specifically designed for handling medical data streams.

BioStream, by HP Laboratories Cambridge, is a real-time, operator-based software solution for managing physiological sensor streams. It is built on top of a general purpose stream processing software architecture. The system processes data using plug-in analysis components that can be easily composed into any configuration for different medical domains. Aurora, by MIT, however is claimed to be a new system for managing data streams and for monitoring applications. The new element is the part of the software system that processes and reacts to continual inputs from many data sources of monitoring sensors. Essentially Aurora is a new database management system designed with a data model and system architecture that embraces a detailed set of stream-oriented operators.

From the literature review, it is apparent that research endeavor has been geared towards the direction of analyzing stream data, tapping the benefits of processing the physiological signals in real-time, and architecting framework of real-time stream-based software system. In 2012, Lin in his book chapter [11] discussed the state of the art and modern research trends of rt-CDSS; specifically he proposed a web-based rt-CDSS with a full architecture showing all the model-view-controller components. In-depth discussions are reported from process scheduling, system integration, to a full networked infrastructure. It is therefore evident that real-time decision system is drawing attentions from both industry and academia, although the details of the analyzer component is still lacking. In [10] we advocated that the main piece of an effective rt-CDSS is an incremental learning model. By far there is no study dedicated to investigate the classifiers for handling data streams in rt-CDSS, to the best of the authors' knowledge. This paper is intended to fill this missing piece.

3. Predicting Future Cases: Problem Definition

As a case study of evaluating the performance of several types of classifiers to be used in rt-CDSS, a diabetes therapy is used. The basis of the diabetes therapy is to replace the lack of insulin by regular exogenous insulin infusion with a right dosage each time, for keeping the patients alive. However, maintaining the blood glucose levels in check via exogenous insulin injection is a tricky and challenging task. Despite the fact that the reactions of human bodies to exogenous insulin vary, the concentration of blood glucose can potentially be influenced by many variables too [1]. These variables include but are not limited to, BMI, mental conditions, hormonal secretion, physical well-being, diets, and lifestyles. Their effects make a synthetic glucose regulation process in diabetic patients highly complex as the bodily reaction to insulin and other factors differs from one person to another. It is all about a matter of a right dosage and the right timing of insulin administration, for regulating the fluctuation of blood glucose concentration at a constant level. Hyperglycemia can occur when the blood glucose level stays chronic above 125 mg/dL over a prolonged period of time. The damages are on different parts of the body, such as stroke, heart attack, erectile dysfunction, blurred vision, and skin infections, just to name a few. At the other end, hypoglycemia occurs when the content of glucose ever falls below 72 mg/dL. Even for a short period of time, hypoglycemia can develop into unpleasant sensations like dysphoria and dizziness and sometimes life-threatening situations like coma, seizures, brain damage, or even death. The challenge now is to try to adopt a classifier which incrementally learns the pattern of a patient's insulin intakes and predicts his blood glucose level in the near future. Should there be any predicted outcome that falls beyond the normal ranges, the rt-CDSS should give a remedy recommendation.

3.1. Data Description. The data used in this experiment are the empirical dataset from AAAI Spring Symposium on Interpreting Clinical Data (http://www.aaai.org/Press/Reports/Symposia/Spring/ss-94-01.php). This data represents a typical flow of measurement records that would be found in any insulin therapy management. The live data feed can serve as an input source for rt-CDSS for the sake of forecasting the condition of the patient in the near future as well as offering medical advice if necessary. The insulin-dependent diabetes mellitus (IDDM) data are event-oriented data because the data is a temporal series of events. Typically there are three groups of events in an insulin therapy, blood glucose measurement (both before/after meals and ad hoc), insulin injections (of different types), and amount of physical exercises. The events are time stamped. However, there is no rigid regularity on how often each of these events would happen. A rough cyclical pattern can be however observed that goes by spacing the insulin injections, probably several times over a day, and the corresponding cycle of blood glucose fluctuation follows closely. These cycles loop over day after day, without specifying the exact timing of each event. One can approximately observe that an average of three or four injections are being applied.

In Figure 2, a sample of these repetitive cycles of events is shown for illustrating the synchronized events. Events of insulin injections and blood glucose measurements are more or less interleaved loosely periodically over time; exercises and sometimes hypoglycemia occur occasionally. In the example presented in Figure 2, two views are provided. The 4-months adaption of insulin injection shows a relatively long-term pattern over time (Figure 2(a)); two exceptionally high doses of insulin over units of 100 were given; more importantly the insulin pattern is never periodically exact, although some cycles are seen to be repeated [23]. The overall insulin intake looks increasing over time from the initial month to the last month. Some events of hypoglycemia have occurred too, sporadically, as represented by red dots in the graph. Zoomed-in views are shown in Figures 2(b) and 2(c), where the timing of the insulin injections are clearly seen. Though the insulin injections are repeating over time, the exact times of injections are seldom the same for any two injections. Sometimes, neutral protamine Hagedorn (NPH) and regular types of injections are taken at the same time. Figure 3(a) shows a change of habit in blood glucose measurements; the frequency has reduced across fifty days by dropping the prelunch and presupper measurements. Figures 3(b) and 3(c) show the same but in time scales of 7 days and 3 days, respectively. The graphs demonstrate a fact that the patterns of timing and doses of insulin injections are aperiodic that elicits substantial computational challenges in testing the classifiers.

3.2. Prediction Assumptions. In order to engineer an effective real-time clinical decision support system, we should use a classification algorithm that can analyse data efficiently and accurately. Traditional decision tree may be a good choice; however, it cannot handle continuous rapid data. To alleviate this problem, incremental classification algorithm, such as VFDT, should be used. For easy illustration when it comes to describing the system processes and workflows throughout this paper, the term VFDT is used that generalized the category of incremental learning methods. In fact, however, other algorithms can be exchanged. Different incremental classifiers in the rt-CDSS model can be adopted.

The prediction is rolling as time passes by. The initial model construction takes about a small portion of the initial data after which the classifier learns and predicts at the same time. One can imagine that there is a time window of 24 hours; when new data rolls in, the old data are flushed out from the memory of the classifier. This way, the classifier can be adaptive to the most current situation and will keep its effectiveness in real time all the time. Regardless of the total size of the data which potentially amount to infinity, the rt-CDSS which is empowered by the incremental learning classifier will still work fine. So in our design, a changing period of 24 hours would be covered for both events that have already happened and will likely happen. Within this period, the classifier continually analyses and remembers the causal relationship between the happened events and the future events. As a case study, the classifier is made to predict future blood glucose level, given the events of insulin injections, meals, and historical blood glucose levels as they all carry certain effects predicting future blood glucose level. The concept of the sliding time window is shown in Figure 4.

As we know, a blood glucose measurement is taken; the measured value is affected by a composite of events that happened during the last several hours. The event may be a meal, an exercise, or an insulin injection. In the design of our experiment, we consider the events which happened during the last 24 hours before the last prediction time point. There are 3 kinds of insulin injections given in the dataset, they are regular insulin, NPH insulin and Ultralente insulin. Regular insulin has at most 6 hours duration effect, NPH has at most 14 hours duration effect, and Ultralente insulin has 24 hours duration effect. Once the prediction point is passed, another fresh set of 24-hours-long events series (24 hours before the previous prediction time point) is loaded to the classifier. This event series include two parts, one is happened event; this part will be extracted from the collected data feed from the monitoring device of the system. For example, assume now that the time is 10:00 we want to predict the blood glucose level at 17:00. Then the system will extract the events data list from yesterday 19:00 to today 10:00 (now), and from the averaged historic record patterns we infer what events the patient would most like to part take in the next 7 hours (from 10:00 to 17:00), such as lunch, snack and exercise. This is to emulate the lifestyle pattern taking into consideration the causality relation between two consecutive days. Some events like meal, exercise, and regular insulin injection only have short effect duration; for these events we only consider the case in the past 6 hours or 3 hours depending on the effect duration of the insulin.

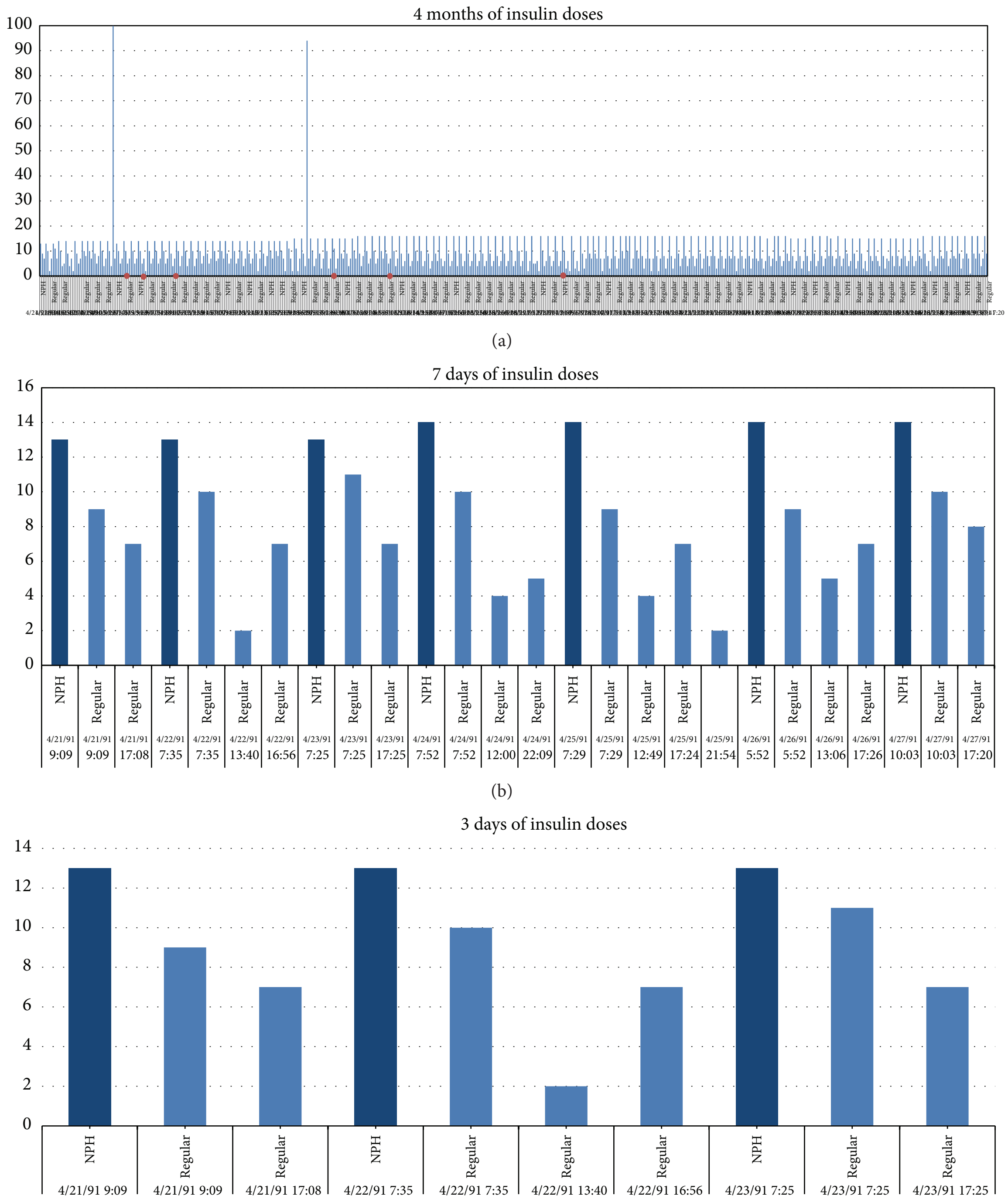
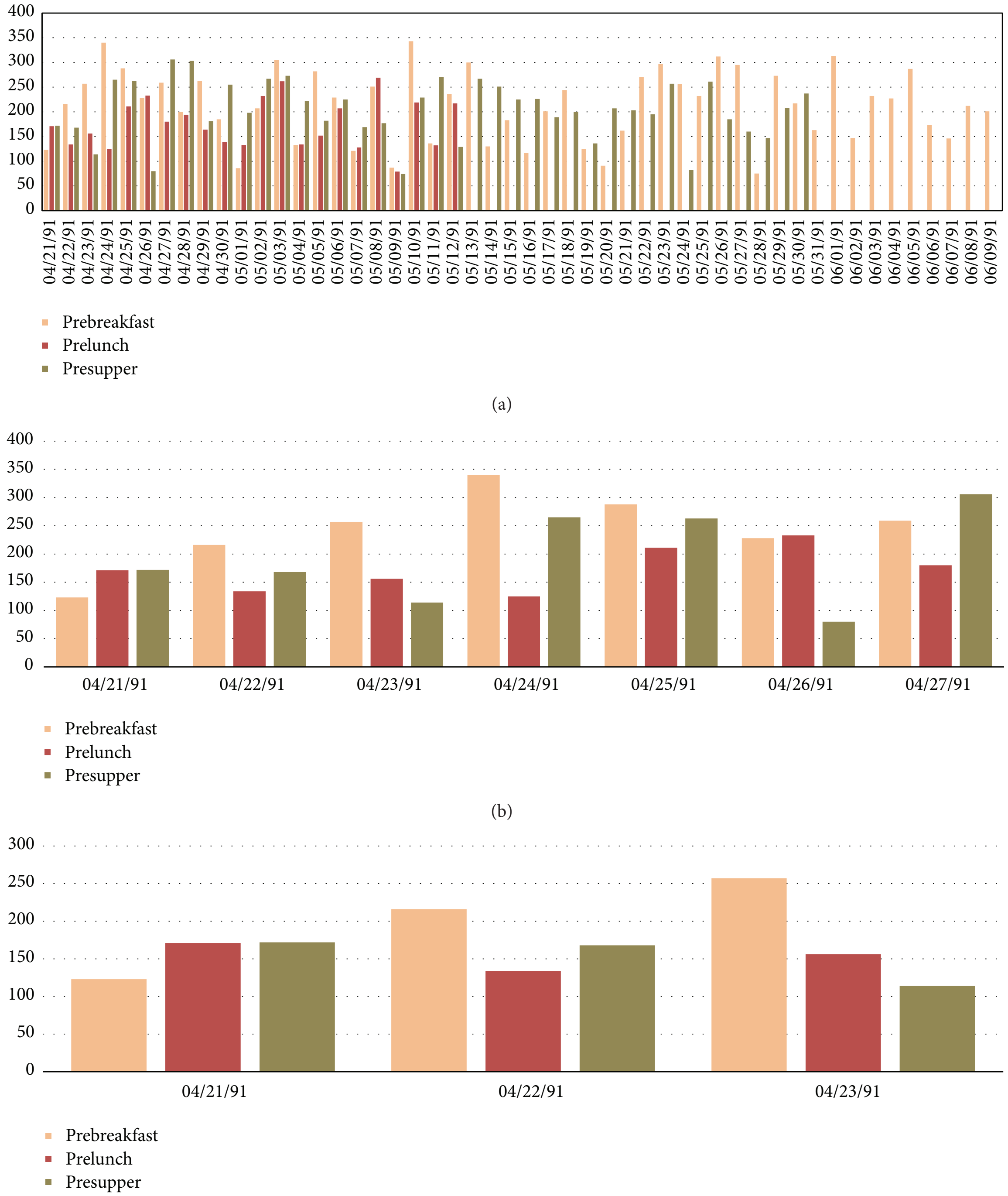


FIGURE 2: Periodic patterns of IDDM events, data taken from a subset of AAAI Spring Symposium on Interpreting Clinical Data. (a) Adaption of insulin for 4 months. (b) Adaption of insulin injections for 7 days. (c) Adaption of insulin injections for 3 days.

3.3. Event List. The data source where the diabetes time-series dataset to be used for our experiment is UCI archive (http://www.ics.uci.edu/~mlearn/) which is popular for benchmarking machine learning algorithms. The events in the diabetes dataset are indexed by numeric codes. Totally there are 20 codes in code list, but not every code is relevant to the blood glucose level which is our predicted target. Some codes are measurements they can provide a blood glucose value and they also represent an event. For example, code 58 represents the event of prebreakfast that means it will happen soon, and it gives the blood glucose count before the breakfast. Code 65 is hypoglycemia symptom that is being measured. The event occurs whenever a measurement of hypoglycemia is detected positive. And there are many



(a)

(b)

(c)

Figure 3: Periodic patterns of blood glucose measurements, data taken from a subset of AAAI Spring Symposium on Interpreting Clinical Data. (a) Time scale of 50 days. (b) Time scale of 7 days. (c) Time scale of 3 days.

different codes that may refer to the same event, such as code 57 and code 48. So we need to simplify the code list and retain only valid events in this list.

From Figure 5 we can see that only four events have effects on the blood glucose levels. The event meal includes several codes, some of them represent a measurement before or after a meal we consider them also representing the time of a meal. For example, when code 58 (with value 100) appears at 9:00, we can know that this person will eat breakfast at nearly 9:00, and the blood glucose before his breakfast is 100. So after simplifying the code list, 4 valid events remain. Each event may have several types. For instance, the event insulin

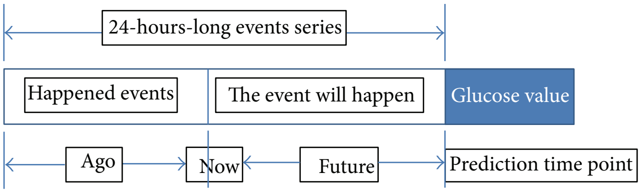


Figure 4: Sliding window for incremental classifier.

dose has 3 types: regular insulin, NPH insulin, and Ultralente insulin. Below is a short list of various types shared by the events.

(i) Event insulin dose: regular insulin, NPH insulin, and Ultralente insulin.

(ii) Event meal: breakfast, lunch, supper, snack, typical meal, more than usual, less than usual.

(iii) Event exercise: typical, more than usual, less than usual.

(iv) Event unspecified special event: exist and N/A.

3.4. The Structure of Training/Testing Instance. All classifiers work on multivariate data which is formatted as an instance of multi-attributed record x_i and it must be described by a set of features $(a_1, a_2, \ldots, a_m)_i$ and a corresponding class label y_i. In this case of diabetes therapy, the data are in time series. A preprocessing software is programmed to convert the events over a time frame of 24 hours into a multiattributed records of m dimensions in n rows (instances).

As described in Section 3.3, the events are filtered so only the relevant event types are used to compose the instances for training and testing. The structure is shown in Figure 6.

According to the general structure specified in Figure 6, a total of 16 attributes would be computed from the event list as follow:

$A0$: measurement code

$A1$: how long ago regular dose

$A2$: how much regular dose

$A3$: how long ago NPH dose

$A4$: how much NPH dose

$A5$: how long ago Ultralente insulin dose

$A6$: how much Ultralente insulin dose

$A7$: the unspecified event in past 6 hours

$A8$: blood glucose level for the previous 3 days

$A9$: hypoglycemia in the past 24 hours

$A10$: last meal in past 6 hours

$A11$: how long ago the last meal in the past 3 hours

$A12$: how long ago the last exercise in the past 24 hours

$A13$: how much exercise

$A14$: Patient ID

$A15$: Blood glucose level (just for training instance).

Table 1: Seven possible target classes.

Target class	BG range (mg/dL)	Limosis	Postprandial
Normal	70~110	Yes	N/A
Abnormal_high	>110	Yes	N/A
Abnormal_low	50~70	Yes	N/A
Normal_1	120~200	No	1 hour
Abnormal_1	50~120 and >200	No	1 hour
Normal_2	70~140	No	2 hours
Abnormal_2	50~70 and >140	No	2 hours

$A8$ is the reference blood glucose level (BGL), which is very important for future blood glucose level prediction. It depends on the BGL in the previous 3 days. From the data analysis we found that there is an important relationship between the current BGL and historical blood glucose level, that exists in the same time period during the previous three days. And we found that the BGL of just one day ago has the most important effect, we call it the factor "1 day before," "2 days before" has second most important effect, and last is "3 days before." So weights of relative importance are arbitrarily set for the 3 factors and $w_1 = 0.5$, $w_2 = 0.3$, $w_3 = 0.2$. $F_1 = 1$ day before, $F_2 = 2$ days before, and $F_3 = 3$ days before. The simple formula that generates the reference BGL, R, is $R = \sum_{i=1}^{3} F_i W_i$ where f is the factor and w is the weight.

3.5. Target Classes. The target class is the prediction result about blood glucose level. Instead of predicting a precise numeric value, the classifier tries to map a new testing instance to one of the 7 classes that describes basically whether the BGL is normal or not. Table 1 shows a class table that illustrates the seven possible normal/abnormal blood glucose levels and their meanings.

As we all know that the blood glucose level will rise up after meals, and it will return to normal level after about 3 hours. So we need to consider the event meal in only the past 3 hours when we do the prediction. In normal situation, one hour postprandial BGL is ranging from 120 to 200 mg/dL (Normal_1) and 2 hours postprandial BG level is ranging from 70 to 140 mg/dL (Normal_2).

4. Experiment

4.1. Experimental Environment and Design. The software system prototype of the rt-CDSS including the classifier is built by Java programming language. The system makes external application-interface calls to the classification algorithms provided by Massive Online Analysis (MOA) (http://moa.cms.waikato.ac.nz). The operating system is MS-Windows 7, 64 bits edition, and the processor is Intel i7 2670 QM 2.20 GHz.

There are 70 diabetes records in our dataset that are collected from 70 different real patients. Each record covers several weeks' to months' diabetes data. We divide every record into two parts; one represents the historical medical data for training and the other part represents future medical data for testing. We use the first part to train the system

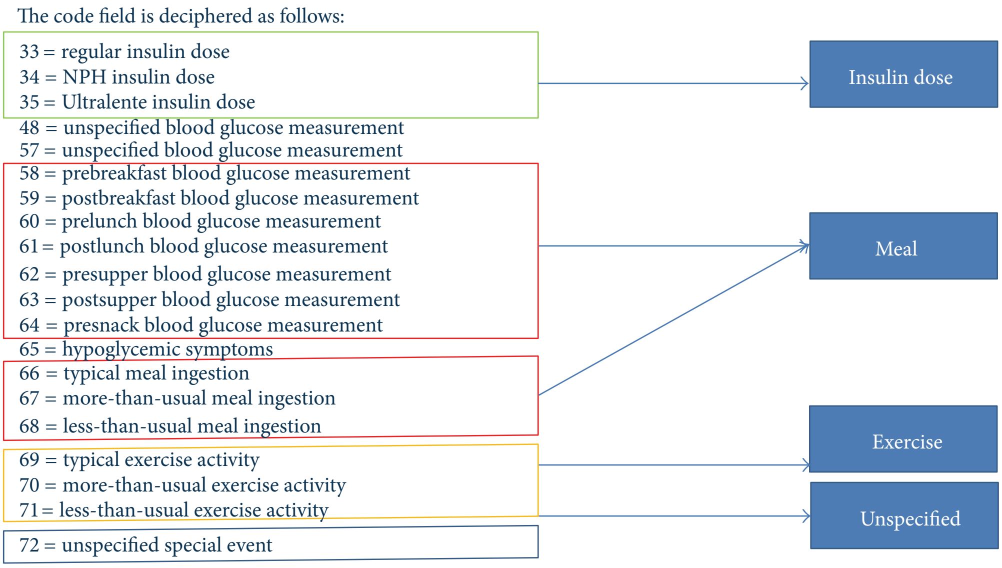

The code field is deciphered as follows:
33 = regular insulin dose
34 = NPH insulin dose
35 = Ultralente insulin dose
48 = unspecified blood glucose measurement
57 = unspecified blood glucose measurement
58 = prebreakfast blood glucose measurement
59 = postbreakfast blood glucose measurement
60 = prelunch blood glucose measurement
61 = postlunch blood glucose measurement
62 = presupper blood glucose measurement
63 = postsupper blood glucose measurement
64 = presnack blood glucose measurement
65 = hypoglycemic symptoms
66 = typical meal ingestion
67 = more-than-usual meal ingestion
68 = less-than-usual meal ingestion
69 = typical exercise activity
70 = more-than-usual exercise activity
71 = less-than-usual exercise activity
72 = unspecified special event
Insulin dose
Meal
Exercise
Unspecified

Figure 5: An event list describing the events by codes.

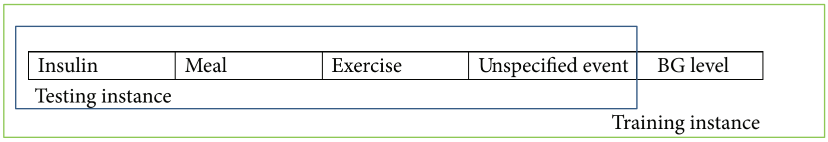


Figure 6: Data instance structure for training/testing a classifier.

with incremental classification algorithms, and we use the second part to do the accuracy test. In reality, when using the system to do a prediction for a new patient, the patient's historical medical record would be loaded in beforehand for initial boot-up training. The historical medical record can be of length of several days (or weeks) of diabetes events. In our experiment, we save the first 1% records from each record as the boot-up training data set.

Firstly, we will conduct the accuracy test for VFDT, iOVFDT [24], Bayes [25], and Perceptron (which is a classical implementation of ANN) [26], respectively. Default parameters are assumed. Then we will analyses their accuracy performance and from there we choose the qualified algorithms for further consistency testes. Finally, we will determine which algorithms work best in our rt-CDSS environment.

4.2. Accuracy Test. All the 70 original patients' records that are available from the dataset would be used for the accuracy test. There are 70 independent accuracy tests. Every record is tested individually using the candidate classifiers and their accuracies are measured, by considering the past 24 hours window of data as training instances, and the testing starts from the first day of the data monitoring till the last. The 70 records are run in sequential manner for the classifiers. Since each instance carries a predefined BGL label, after running through the full course of prediction, the predicted results could be compared with the actual results. By definition, the accuracy is given as accuracy = (total number of correctly classified instances/the total number of instances available for this particular patient) × 100. The total accuracy is therefore the average of the accuracies over 70 patients' BGL predictions during the course of diabetes therapy. The overall statistics of the accuracy tests are shown in Table 2.

From Table 2, it is observed that the average accuracy for all the candidate algorithms are acceptable except Perceptron. For the algorithms that have acceptable accuracies such as VFDT, iOVFDT, and Bayes, over 75% of the cases they are predicting are at an accuracy higher than or equal to 81%. That means in most situations the rt-CDSS with these qualified algorithms are making useful predictions. For Perceptron, however, during the prediction course of 75% of the records its accuracy is lower than 53.814%, that is just marginally better than random guesses. As a concluding remark, Perceptron fails to adequately predict streaming data when the initial training sample is just about 10%. Thus it is not a suitable candidate algorithm to be used in rt-CDSS when the incoming data stream is dynamic, complex, and irregular.

Figure 7 is a boxplot diagram for comparing visually the performances of the candidate algorithms. Boxplot diagram is an important way to graphically depict groups of numerical data through their quartiles. It is often used as a method to show the quality of a dataset, where in this case the performance results of it.

From the boxplot, we can see that the performances between VFDT and iOVFDT are so close; their accuracy

Table 2: Results of the accuracy test.

Accuracy	VFDT	iOVFDT	Bayes	Perceptron
Mean	87.4314%	86.4102%	82.6453%	25.4779%
Max	95.6810%	93.7930%	95.1720%	91.6670%
Min	78.8460%	79.3100%	25.3010%	0.0000%
Std. dev.	0.0403	0.0342	0.1184	0.3164
Quartiles 25	85.4633	84.4560	81.5950	0.0000
Quartiles 50	87.3395	86.1990	85.6170	11.4720
Quartiles 75	90.2530	89.0150	88.4955	53.8140

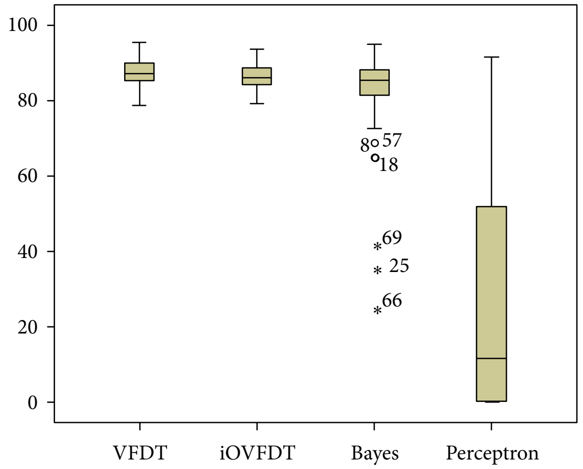


Figure 7: Boxplot diagram of accuracy performances for the classifiers.

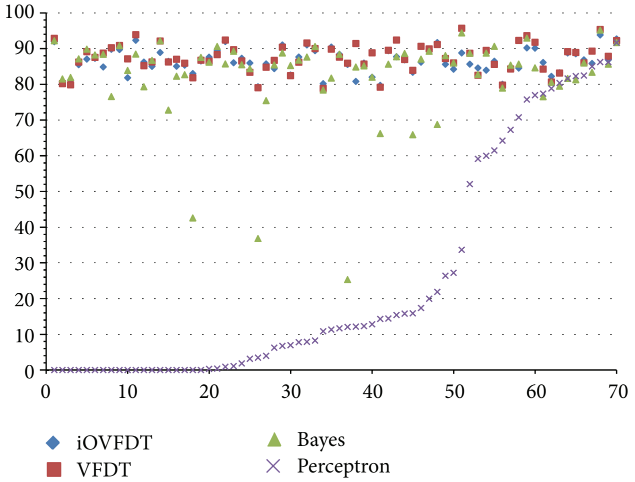


Figure 8: Scatterplot diagram of accuracy performances for the classifiers.

distributions are very similar, and there is no outlier in their distributions. The maximum accuracy for iOVFDT is slightly lower than that of VFDT, but iOVFDT has an overall consistent accuracy performance and a higher minimum accuracy compared to VFDT. That is because iOVFDT was designed to achieve optimal balance of performance, where the result may not be maximum but well balanced in consideration of the overall performance.

For Bayes algorithm the accuracy is basically acceptable, but there are 3 outliers. These extreme values are associated with records 69, 25, and 66, where the accuracies fall below 50%. It means Bayes works well for most of the records, but there also exist some situations where Bayes fails to predict accurately. The worst performance as seen from the boxplots is by Perceptron; in most cases, it predicts incorrectly.

The scatter plot as depicted in Figure 8 shows an interesting phenomenon when the accuracy results are viewed longitudinally across the whole course of prediction in rt-CDSS. The qualified classifiers such as VFDT, iOVFDT and Bayes are all able to start showing early high accuracies especially for VFDT and iOVFDT. They are able to maintain this high level of accuracies across the full course at over >80%. The performance for Bayes is also quite stable starting from the initial record to the end, except several outlier points.

In contrast, Perceptron picked up the accuracy rate after being trained with approximately 25 sets of patients' records; the accuracy trend increases gradually over the remaining records and climbs up high on par with the other classifiers near the end. In fact, its maximum accuracy rate is 91.667%, while the other prediction accuracies for the other classifiers range from 93.793% to 95.681%. And the accuracy for Perceptron algorithm seems to be able to further increase should the provision of training data be continued. This implies that Perceptron algorithm is capable of delivering good prediction accuracy, but under the condition that sufficient training data must be made available for inducing a stable model. However, in scenario of real-time data stream in which rt-CDSS is embracing, incremental learning algorithms have their edge in performance.

Overall, with respect to accuracy, the best performers are VFDT and iOVFDT. The performance for Bayes is acceptable though outliers occur at times. Given the fact that Perceptron is unable to achieve an acceptable level of accuracy in the initial stage of incremental learning, it is dropped from further tests in our rt-CDSS simulation experiment. The remaining qualified algorithms are then subject to further tests.

4.3. Consistency Test. Kappa statistics is used for testing the consistency of accuracies achieved by each of the VFDT,

Table 3: The Kappa statistics for the candidate classifiers.

Algorithm	VFDT	iOVFDT	Bayes
Kappa statistics	0.587	0.605	0.678
Reference		Remarks	
0.0~0.20		Slight	
0.21~0.40		Fair	
0.41~0.60		Moderate	
0.61~0.80		Substantial	
0.81~1		Almost perfect	

iOVFDT, and Bayes classifiers. Kappa statistics is generally used in data mining, statistical analysis, and even assessment of medical diagnostic tests [27], as an indicator on how "reliable" a trained model is. It basically reflects how consistent the evaluation results obtained from multiple interobservers are and how well they are agreed upon. A full description of the Kappa statistics can be found in [28]. Generally a Kappa of 0 indicates that agreement is equivalent to chance, where as a Kappa of 1 means perfect agreement. It loosely defines here as a measure of consistency by saying a model that has a high Kappa value is a consistent model that would expect about the same level of performance (in this case, accuracy) even when it is tested with datasets from other sources. The Kappa statistics is computed from the 70 patients' records via a 10-fold cross-validation with each fold of different combination of partitions (training and testing) as different inter-observers, randomly picked from the whole dataset.

The definition of Kappa statistic is defined as $K = (Po - Pc)/(1 - Pc)$, where Po is the observed agreement and Pc is chance agreement. The results of the Kappa statistics from the candidate classifiers are tabulated in Table 3.

We can see from Table 3 that the Bayes classifier has the highest consistency value relatively; it belongs to the substantial group of Kappa statistics. The other 2 algorithms are located in the moderate group. The result shows that all the three algorithms have considerably moderate and substantial consistency in rt-CDSS. Higher Kappa statistics are yet to be obtained probably due to the irregularity of events in the datasets and of the 70 patients the diabetes therapy patterns vary a lot.

4.4. Test of ROC Curve and AUC. ROC is an acronym for Receiver Operating Characteristic; it is an important means to evaluate the performance of a binary classifier system. It is created by plotting the fraction of true positives out of the positives (TP = true positive rate) in x-axis and the fraction of false positives out of the negatives (FP = false positive rate) in y-axis. The terms positive and negative describe the classifier's prediction results, and the terms true and false refer to whether the prediction results correspond to the fact or not. The standard contingency table or confusion matrix for binary classification is shown in Table 4.

In our rt-CDSS, the classifiers are multiclasses classifiers rather than binary classifiers as they predict the future conditions of the patients into one of the seven BGLs. So we need a modification to extend the conventional ROC curve in our evaluation experiment. By following the modification which was reported in [29], a binary ROC curve is extended for the use for multiclasses classifiers. The modified contingency table is shown in Table 5.

The occurrences for TP and FP for each class are counted respectively. A small assumption is made during the counting: when counting for a class, an instance in this class is counted as "yes" and the instances in other class are "no", just like binary classifier. Then by adding up all the TPs as total $\mathrm{TP}_{\mathrm{multi}}$ and all the FPs as total $\mathrm{FP}_{\mathrm{multi}}$, we compute the $\mathrm{TP}_{\mathrm{multi}}$ and $\mathrm{FP}_{\mathrm{multi}}$ and derive a composite ROC by calculating sensitivity and specificity as the y-axis and x-axis of the ROC chart accordingly.

Sensitivity is named after $\mathrm{TP}_{\mathrm{multi}}$ which is sometimes called recall rate. It counts about the proportion of actual positives which are identified correctly by the classifier. The proportion here is the percentage of diabetes patients of abnormal BGL who are correctly predicted as having the abnormal condition in the rt-CDSS. Specificity which is the $\mathrm{FP}_{\mathrm{multi}}$ and sometimes known as the true negative rate measures the proportion of negatives which are predicted correctly as such. The proportion is the percentage of diabetes patients with normal BGL who are correctly predicted as not having the abnormal BGL.

Ideally a perfect classifier should be 100% sensitivity and 100% specificity, meaning it can predict that all patients who will have abnormal BGL really will have the condition; patients who will not have the abnormal BGL will actually be free from it. So when plotting sensitivity and specificity on a ROC plot, the curve should be the higher the better in these two directions. Theoretically any classifier will display certain trade-off between these two measures. For example, in rt-CDSS in which the user is testing for extra precaution for health assessment for the diabetes patient, the classifier may be set to consider more thorough life events that may be related to a sudden change in BGL, even though they are minor ones (low specificity), and perhaps higher influential factors are adjusted for these event variables that may directly or indirectly trigger the change in BGL (high sensitivity). This trade-off can be perceived graphically by the shape of the ROC. The ROCs for the classifiers are shown in Figure 9. The corresponding AUC numeric results of the ROCs are tabulated in Table 6.

From the ROC curve and AUC (area under the curve) as shown in Figure 9 and Table 6, we can see that the Bayes classifier has the largest AUC, and the larger an AUC is the better performance it gives. VFDT and iOVFDT have almost the same AUCs. It means their performances in the rt-CDSS model are very close. Perceptron has the smallest AUC in this design, amounts to nearly 0.5; it means that the classifier works almost randomly.

4.5. Test of Precision, Recall and F-Measure. In pattern recognition and data mining, precision is the fraction of relevantly retrieved instances. In the situation of rt-CDSS classifications, precision is a measure of the accuracy provided that a specific class has been predicted. It is calculated by this simple formula: precision = TP/(TP + FP).

Table 4: Contingency table and the remarks.

	Actual class (observation)	
Predicted class (expectation)	TP (true positive)	FP (false positive)
	Correct result	Unexpected result
	FN (false negative)	TN (true negative)
	Missing result	Correct absence of result

Table 5: Contingency table for multiclasses classifiers.

Actual values	Predictions outcomes
TP_1	FN_1
C_{11}	C_{12} C_{13} $\cdots$ C_{1d}
FP_1	TN_1
C_{21}	C_{22} C_{23} $\cdots$ C_{2d}
$\vdots$	$\vdots$ $\vdots$ $\vdots$ $\vdots$
C_{d1}	C_{d2} C_{d3} $\cdots$ C_{dd}

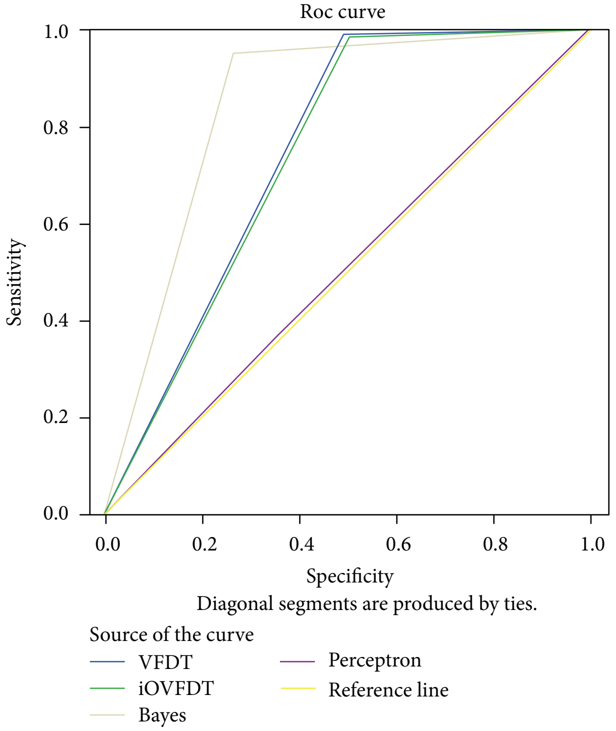


Figure 9: ROC and AUC performances of the classifiers.

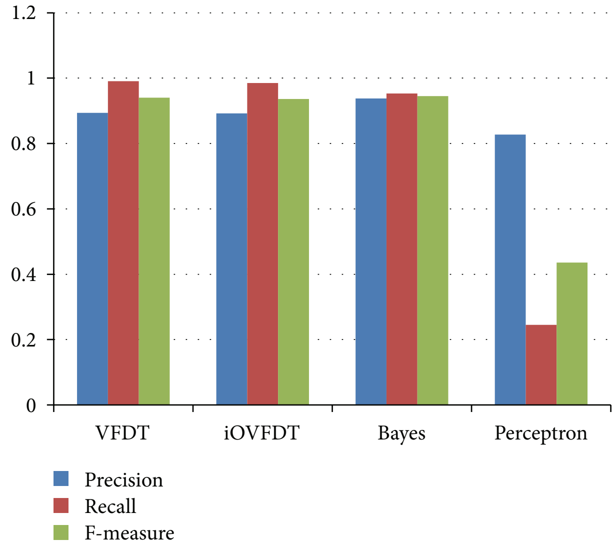


Figure 10: Performances of the classifiers in terms of precision, recall, and F-measure.

Recall is defined as the fraction of relevantly retrieved instances. We can infer that the same part of both precision and recall is relevance, based on which they all make a measurement. Usually, precision and recall scores are not discussed in isolation, and the relationship between them is inverse, indicating that one increases and the other decreases. Recall is defined as recall = TP/(TP + FN).

In a classification task, recall is a criterion of the classification ability of a prediction model to select labeled instances from training and testing datasets. A precision with score 1.0 means that every instance with label belonging to the specific class (predicted by the classifier) does indeed belong to that class in fact. Whereas a recall of score 1.0 means that each instance from that particular class is labeled to this class and all are predicted correctly, none shall be left out.

F-measure is the harmonic mean of precision and recall, that is: F measure $= 2/((1/\text{Precision}) + (1/\text{Recall})) = (2 \cdot \text{Precision} \cdot \text{Recall})/(\text{Precision} + \text{Recall})$. It is also known as balanced F score or F-measure in tradition, because recall and precision are equally weighted. The general formula for F_β measure is $F_\beta = (1 + \beta^2)/((1/\text{Precision}) + (\beta^2/\text{Recall})) = ((1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall})/(\beta^2 \cdot \text{Precision} + \text{Recall})$. As mentioned before, precision and recall scores should be taken into account simultaneously because they have a strong relation essentially. Consequentially, both are combined into a single measure, which is F-measure, which is perceived as a well-rounded performance evaluation, more highly valued than the simple accuracy.

The performance results of precision, recall, and F-measure are then tabulated in Table 7 and shown in bar-chart in Figure 10.

With respect to precision value, we can see Bayes has the highest. The precision values between VFDT and iOVFDT are nearly identical. There is a strange observation that the precision score for Perceptron is also quite high (0.827), despite the fact that Perception was most down rated in the accuracy test. This phenomenon can be explained that

Table 6: Numeric results of AUCs of the classifiers.

Test result Variable(s)	Area	Std. error[a]	Asymptotic sig.[b]	Asymptotic 95% confidence interval	
				Lower bound	Upper bound
VFDT	0.752	0.007	0	0.739	0.765
iOVFDT	0.745	0.007	0	0.732	0.758
Bayes	0.847	0.005	0	0.836	0.858
Perceptron	0.508	0.007	0.228	0.495	0.521

The test result variable(s): VFDT, iOVFDT, Bayes, and Perceptron has at least one tie between the positive actual state group and the negative actual state group. Statistics may be biased.

[a] Under the nonparametric assumption.

[b] Null hypothesis: true area = 0.5.

Table 7: Numeric results of AUCs of the classifiers.

Classifier	VFDT	iOVFDT	Bayes	Perceptron
Precision	0.894	0.892	0.938*	0.827
Recall	0.991*	0.985	0.953	0.245
F-measure	0.940	0.936	0.945*	0.436

*Refers to a winning classifier that has the highest performance.

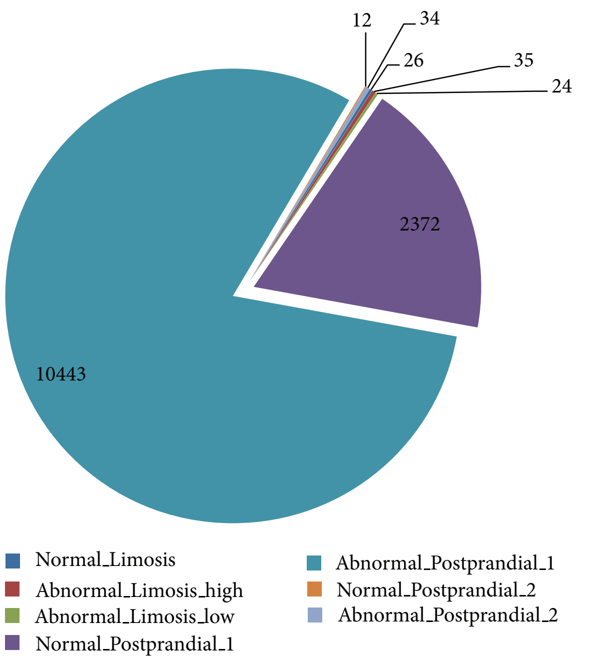


Figure 11: Distribution of the instances among the target classes.

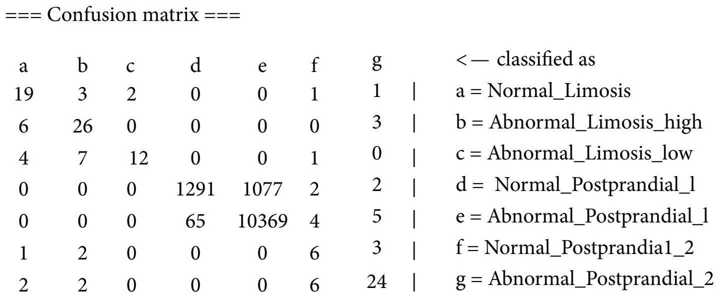
Perceptron rarely makes a positive decision; from the histogram above in Figure 10 we can easily see that the total sums of TP and FP for perceptron is much less than the others. But out of these rare predictions, Perceptron has a relatively high rate in precision.

When it comes to recall criterion, VFDT has the best score, iOVFDT and Bayes both have good Recall values (>0.95). We can see quite clearly that Perceptron made a lot of false negative prediction, as its Recall value is only 0.245. This is an immature sign of its underlying model is under-trained with insufficient training samples.

For the final composite scores, *F*-measures, as shown in Table 7, Bayes outperforms the rest of the others. The candidate that has the second highest *F*-measure is VFDT whose difference is merely 0.005. In summary, Bayes classifier can be a good candidate for implementing rt-CDSS given the fact that it overall outperforms the rest in Precision, *F*-measure, AUC and Kappa statistics. However one drawback is the outlier predictions that can occur at Bayes classifier though seldom in the course of prediction. In medical applications, such anomaly in performance can lead to grave consequences. The underperformance of Bayes may be due to a large amount of conflicting conditions in the dataset where a particular class out of the seven classes is highly unbalanced (biased). As shown in Figure 11, the class called *Abnormal_Postprandial_1* has an unusually high number of instances (10,443) compared to the rest of the classes.

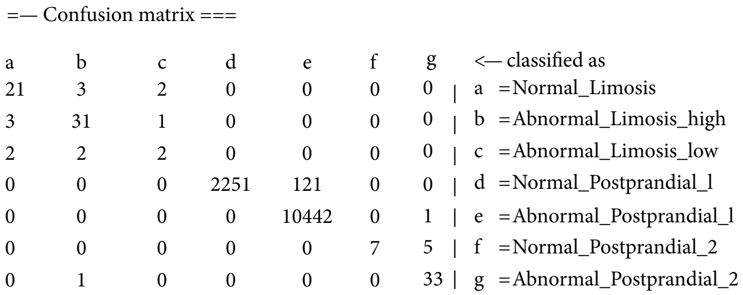
By comparing the confusion matrices of Bayes and OVFDT, as shown in Figures 12 and 13, respectively, one can observe the reason behind the shortcoming of Bayes prediction. In the biased class which dominates most of the training instances, Bayes incorrectly classified 1,077 instances pertaining to *Abnormal_Postprandial_1* compared to OVFDT which classified wrongly of 121 for the same class. This particular inaccuracy at the biased class rated down the performance of Bayes as a whole given its somewhat rigid probabilistic network. On the other hand, decision tree type of classifiers such as OVFDT and iOVFDT are about to grow extra decision paths to relieve this specific inaccuracy hotspot.

iOVFDT is the second best in Kappa statistics, and provides reasonably well performance in the other measures (though not the highest). It could be an appropriate choice in rt-CDSS given its stable performance considering all aspects of evaluation. Perception is unsuitable for classifying data stream.



```
=== Confusion matrix ===

 a    b    c     d      e     f    g        <— classified as
19    3    2     0      0     1    1   |    a = Normal_Limosis
 6   26    0     0      0     0    3   |    b = Abnormal_Limosis_high
 4    7   12     0      0     1    0   |    c = Abnormal_Limosis_low
 0    0    0   1291   1077    2    2   |    d =  Normal_Postprandial_l
 0    0    0    65   10369    4    5   |    e = Abnormal_Postprandial_l
 1    2    0     0      0     6    3   |    f = Normal_Postprandia1_2
 2    2    0     0      0     6   24   |    g = Abnormal_Postprandial_2
```

Figure 12: Confusion matrix of Bayes classifier.



```
=— Confusion matrix ===

a     b     c     d      e      f    g     <— classified as
21    3     2     0      0      0    0  |  a  =Normal_Limosis
3     31    1     0      0      0    0  |  b  =Abnormal_Limosis_high
2     2     2     0      0      0    0  |  c  =Abnormal_Limosis_low
0     0     0    2251   121     0    0  |  d  =Normal_Postprandial_l
0     0     0     0    10442    0    1  |  e  =Abnormal_Postprandial_l
0     0     0     0      0      7    5  |  f  =Normal_Postprandial_2
0     1     0     0      0      0   33  |  g  =Abnormal_Postprandial_2
```

Figure 13: Confusion matrix of OVFDT classifier.

5. Conclusion and Future Works

Clinical decision support system (CDSS) has drawn considerate attentions from researchers from information technology discipline as well as medical practitioners. This is a sequel paper which follows a new novel design of real-time clinical decision support system (rt-CDSS) with data stream mining. In our previous paper, a conceptual framework has been proposed. However, one important internal process which is the core of the whole system is the classifier which is supposed to predict the future condition of a patient based on his past historic events as well as other generalized medical propensity information. Once a prediction is made, the leaf pointers of the class nodes of the decision tree will fetch the relevant prescribed medical guidelines for recommendation. It can be understood that such classifier inside the rt-CDSS would need to possess the following capabilities: (1) handling data stream such as live feeds of biosignals monitoring devices, other instant measurements of vital signs, and physiological reactions/responses to drugs treatments; (2) a very short time delay in model updates when new data arrives; and perhaps most importantly (3) accurate and consistent prediction performance.

Traditional classifiers which have been widely used in CDSS and whose designs based on structured electronic patients' records (instead of stream data) are known to come short of satisfying the three requirements. The main distinction between traditional CDSS and rt-CDSS is the reaction time required; CDSS is centered on disease that has certain length of onset time and rt-CDSS is for emergency medical situations; hence timely and accurate decisions are very crucial. It was already studied in the other papers that traditional classifiers require a complete scanning of a full training dataset every time a new piece of data is added on. Such batch-based learning is not efficient enough to learn and adapt to fast moving data stream in real-time. rt-CDSS is a new breed of decision support tools. To the best of the authors' knowledge, none of the related works has investigated the issue of finding a suitable classifier for rt-CDSS. This paper contributes to a performance evaluation of several incremental learning algorithms together with an artificial neural network algorithm that has been used extensively for traditional CDSS. A case study of diabetes therapy with real patents' data was used in the evaluation experiment which simulates a therapeutic decision-support scenario where real-time blood glucose level is predicted based on various insulin intakes and life-time events.

Our results show that classifier of artificial neural network gives unsatisfactory performance under a rolling sequence of event data. A neural network usually needs to be sufficiently trained by the full volume of dataset which may not be available in data streaming environment. Bayes algorithm is found to be having the highest consistency in terms of Kappa statistics and few other performance scores; its prediction is stained with some outliers (sudden accuracy degradations) in the course of prediction. VFDT on the other hand has the highest accuracy, but its accuracy for one dataset may not always be as consistent as that for another dataset, when compared to iOVFDT whose performance is rather stable.

As future works, the authors are inclined to test a wider range of stream mining algorithms that are available in the literature. The same performance testing would be repeated while the classifiers are to be integrated with the other components of the rt-CDSS and be tested as a whole system. Scenario and dataset of higher complexity should be tested with the classifiers too, for example, ICU data where multiple data feeds (ECG, respiratory measures, blood pressure, oxygen in blood, etc.) are streaming into the rt-CDSS in real time.

Acknowledgment

The authors are thankful for the financial support from the research grant "Adaptive OVFDT with Incremental Pruning and ROC Corrective Learning for Data Stream Mining," Grant no. MYRG073(Y2-L2)-FST12-FCC, offered by the University of Macau, FST, and RDAO.

References

[1] A. Berlin, M. Sorani, and I. Sim, "A taxonomic description of computer-based clinical decision support systems," *Journal of Biomedical Informatics*, vol. 39, no. 6, pp. 656–667, 2006.

[2] O. S. Mohammed and R. Benlamri, "Building a diseases symptoms ontology for medical diagnosis: an integrative approach," in *Proceedings of the IEEE International Conference on Future Generation Communication Technology (FGCT '12)*, pp. 104–108, British Computer Society, London, UK, December 2012.

[3] J. Fiaidhi and S. Mohammed, "Adopting personal learning environments for sharing electronic healthcare records," in *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 4011–4016, 2010.

[4] S. Fong, Y. Hang, S. Mohammed, and J. Fiaidhi, "Stream-based biomedical classification algorithms for analyzing biosignals," *Journal of Information Processing Systems*, vol. 7, no. 4, pp. 717–732, 2011.

[5] I. T. BjØrk and G. A. Hamilton, "Clinical decision making of nurses working in hospital settings," *Nursing Research and Practice*, vol. 2011, Article ID 524918, 8 pages, 2011.

[6] S. Fong and Y. Hang, "Enabling real-time business intelligence by stream data mining," in *New Fundamental Technologies in Data Mining*, K. Funatsu, Ed., Intech, Vienna, Austria, 2011.

[7] Y. Hang and S. Fong, "An experimental comparison of decision trees in traditional data mining and data stream mining," in *Proceedings of the 6th International Conference on Advanced Information Management and Service (IMS '10)*, pp. 442–447, Seoul, Korea, December 2010.

[8] M. J. Yuan, "Watson and healthcare: how natural language processing and semantic search could revolutionize clinical decision support," Tech. Rep., IBM Developer Works, 2011.

[9] J. Sun, D. Sow, J. Hu, and S. Ebadollahi, "A system for mining temporal physiological data streams for advanced prognostic decision support," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM '10)*, pp. 1061–1066, Sydney, Australia, December 2010.

[10] Y. Zhang, S. Fong, J. Fiaidhi, and S. Mohammed, "Real-time clinical decision support system with data stream mining," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 580186, 8 pages, 2012.

[11] H.-C. Lin, "Real-time clinical decision support system," in *Medical Informatics*, pp. 111–136, InTech, 2012.

[12] S. Fong, S. Mohammed, J. Fiaidhi, and C. K. Kwoh, "Using causality modeling and Fuzzy Lattice Reasoning algorithm for predicting blood glucose," *Expert Systems With Applications*, vol. 40, no. 18, pp. 7354–7366, 2013.

[13] V. Patkar, D. Acosta, T. Davidson, A. Jones, J. Fox, and M. Keshtgar, "Cancer multidisciplinary team meetings: evidence, challenges, and the role of clinical decision support technology," *International Journal of Breast Cancer*, vol. 2011, Article ID 831605, 7 pages, 2011.

[14] J. Szkoa, K. Pancerz, and J. Warcho, "Recurrent neural networks in computer-based clinical decision support for laryngopathies: an experimental study," *Computational Intelligence and Neuroscience*, vol. 2011, Article ID 289398, 8 pages, 2011.

[15] P. Walsh, P. Cunningham, S. J. Rothenberg, S. O'Doherty, H. Hoey, and R. Healy, "An artificial neural network ensemble to predict disposition and length of stay in children presenting with bronchiolitis," *European Journal of Emergency Medicine*, vol. 11, no. 5, pp. 259–264, 2004.

[16] L. B. Gerald, S. Tang, F. Bruce et al., "A decision tree for tuberculosis contact investigation," *American Journal of Respiratory and Critical Care Medicine*, vol. 166, no. 8, pp. 1122–1127, 2002.

[17] B. G. Buchanan and E. H. Shortliffe, *Rule Based Expert Systems: the MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, USA, 1984.

[18] Skepticsm about MYCIN Method, June 2013, http://en.wikipedia.org/wiki/Mycin#Method.

[19] H. R. Warner Jr. and O. Bouhaddou, "Innovation review: iliad—a medical diagnostic support program," *Topics in Health Information Management*, vol. 14, no. 4, pp. 51–58, 1994.

[20] A. Bar-Or, D. Goddeau, J. Healey, L. Kontothanassis, and B. Logan, "BioStream: a system architecture for real-time processing of physiological signals," in *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference (EMBS '04)*, pp. 3101–3104, San Francisco, Calif, USA, September 2004.

[21] D. J. Abadi, D. Carney, U. Çetintemel et al., "Aurora: a new model and architecture for data stream management," *The VLDB Journal*, vol. 12, no. 2, pp. 120–139, 2003.

[22] B. P. Kovatchev, "Diabetes technology: markers, monitoring, assessment, and control of blood glucose fluctuations in diabetes," *Scientifica*, vol. 2012, Article ID 283821, 14 pages, 2012.

[23] D. Takahashi, Y. Xiao, and F. Hu, "A survey of insulin-dependent diabetes—part II: control methods," *International Journal of Telemedicine and Applications*, vol. 2008, Article ID 739385, 14 pages, 2008.

[24] H. Yang and S. Fong, "Incremental optimization mechanism for constructing a decision tree in data stream mining," *Mathematical Problems in Engineering*, vol. 2013, Article ID 580397, 14 pages, 2013.

[25] T. Anwar, S. Asghar, and S. Fong, "Bayesian based subgroup discovery," in *Proceedings of the 6th International Conference on Digital Information Management (ICDIM '11)*, pp. 154–161, Melbourne, Australia, September 2011.

[26] G. Robertson, E. D. Lehmann, W. Sandham, and D. Hamilton, "Blood glucose prediction using artificial neural networks trained with the AIDA diabetes simulator: a proof-of-concept pilot study," *Journal of Electrical and Computer Engineering*, vol. 2011, Article ID 681786, 11 pages, 2011.

[27] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: the kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–363, 2005.

[28] S. Fong and A. Cerone, "Attribute overlap minimization and outlier elimination as dimensionality reduction techniques for text classification algorithms," *Journal of Emerging Technologies in Web Intelligence*, vol. 4, no. 3, pp. 259–263, 2012.

[29] H. Yang, S. Fong, R. Wong, and G. Sun, "Optimizing classification decision trees by using weighted naïve bayes predictors to reduce the imbalanced class problem in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 460641, 15 pages, 2013.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 197925, 11 pages
http://dx.doi.org/10.1155/2013/197925

Research Article

New Optical Methods for Liveness Detection on Fingers

Martin Drahansky,[1] Michal Dolezel,[1] Jan Vana,[1] Eva Brezinova,[2] Jaegeol Yim,[3] and Kyubark Shim[4]

[1] *Faculty of Information Technology, Brno University of Technology, Bozetechova 2, 61266 Brno, Czech Republic*
[2] *Department of Dermatology and Venereology, Faculty of Medicine, St. Anne's University Hospital, Masaryk University, Kamenice 753/5, 625 00 Brno, Czech Republic*
[3] *Department of Computer Engineering, Dongguk University at Gyeongju Gyeongbuk, Sekjang-Dong, Gyeong Ju, Gyeongsangbuk-do 780-714, Republic of Korea*
[4] *Department of Statistics and Information Science, Dongguk University at Gyeongju Gyeongbuk, Sekjang-Dong, Gyeong Ju, Gyeongsangbuk-do 780-714, Republic of Korea*

Correspondence should be addressed to Martin Drahansky; drahan@fit.vutbr.cz

Received 4 July 2013; Revised 14 August 2013; Accepted 14 August 2013

Academic Editor: Tai-hoon Kim

This paper is devoted to new optical methods, which are supposed to be used for liveness detection on fingers. First we describe the basics about fake finger use in fingerprint recognition process and the possibilities of liveness detection. Then we continue with introducing three new liveness detection methods, which we developed and tested in the scope of our research activities—the first one is based on measurement of the pulse, the second one on variations of optical characteristics caused by pressure change, and the last one is based on reaction of skin to illumination with different wavelengths. The last part deals with the influence of skin diseases on fingerprint recognition, especially on liveness detection.

1. Introduction

A first phase of fingerprint processing is obtaining a digitalized fingerprint [1]. The traditional (dactyloscopic) method uses the ink to get the fingerprint onto a piece of paper. This piece of paper is then scanned using a common (office) scanner, but this method is not interesting for our following description. The modern live fingerprint readers are used—they use different physical effects to acquire the image of a finger.

While the first generation scanners used optical techniques [2], a variety of sensing techniques are used today and almost all of them belong to one of the three families [2]: *optical*, *solid state*, and other (e.g., ultrasound). Solid-state sensors are now gaining great popularity because of their compact size, which facilitates embedding them into laptop computers, cellular phones, smart cards, and so forth.

We can define all fingerprint scanning technologies as separate methods, because they do use special physical effects to obtain the impression of a finger—to the known technologies belong [3–6]: *optical*, *capacitive*, *ultrasound*, *e-field*, *electrooptical*, *pressure sensitive*, *thermal*, and *MEMS* (microelectromechanical systems) [7].

When a legitimate user has already registered his finger in a fingerprint system, there are still several ways how to deceive the system. These ways and methods are well known. Further details can be seen, for example, in [8–10]. In order to deceive the fingerprint system, an attacker may use any of them. In this paper, we will focus only on deceiving the fingerprint system by using an artificial finger (print) fakes.

The term *fake samples* (e.g., *fake finger (print)*) [9, 11, 12] may be widely used with reference to biometric samples which are used to deceive biometric systems. However, the term "artificial samples" (e.g., artificial fingerprint) (see Figure 1) corresponds to biometric samples which are entirely artificially produced. Mathematically, "artificial samples" represent a subset of "fake samples", because the set "fake samples" may also include modifications of live samples [9, 11],

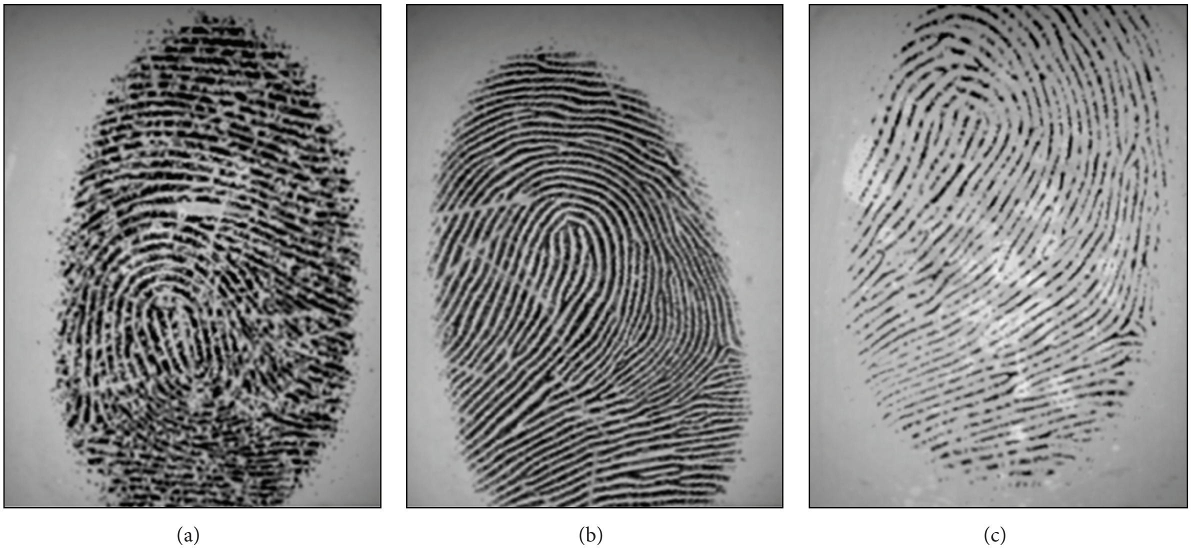


Figure 1: Artificial fingerprints generated by SFinGe [6, 16].

for example, “artificial samples” are fingerprints produced from a mold, but the set “fake samples” contains also injured or otherwise modified live fingers and biometric samples.

The traditional authentication used during the enrollment process should be stronger than all subsequent verifications, because each of the subsequent verifications depends on the strength of authentication and the quality of biometric samples acquired during enrollment. However, in many biometric implementations the enrollment process may become a weak link. One possible protection method could be found in [13] or [14] or [15].

The implications of this susceptibility to *spoofing* [19–21]—defeating a biometric system through fake biometric samples—include the following [22].

(i) Fake finger attacks may be mounted against existing enrollments in order to gain access to a protected facility, computer, or other resource.

(ii) A fake finger may be used for authentication at a given computer or border crossing in order to fraudulently associate an audit trial with an unwitting individual.

(iii) A fake finger may be used to enroll in a biometric system and then be shared across multiple individuals, thereby undermining the entire system.

(iv) An individual may repudiate transactions associated with his account or enrollment—claiming instead that they are the result of attacks—due to the inability of the biometric system to ensure liveness.

Given biometrics’ widespread acceptance [22] as a solution for a range of public and private sector applications such as civil identification, network security, border control, and point of sale authentication, the question of liveness detection in leading biometric technologies must be addressed.

The concept of liveness detection can be framed by considering the detection of liveness versus the detection of nonliveness [22]. The liveness detection may take place at the acquisition stage, such that nonlive data are not acquired, or at the processing stage, such that nonlive data are not processed.

Let us assume that the enrollment process was properly performed and the user has registered his biometric sample, without which it is pointless to assure the system in the verification/identification phase. In general, three basic attacks are feasible whenever the liveness detection is not well implemented [11, 23].

(i) *Presenting artificial samples of the registered user* is the most common and most important threat or risk in this category because of the relative easy effort and variety of ways in which such an attack could be realized. Artificial samples could be produced in many ways from many materials, which poses a problem for detection countermeasures. Dishonest acts with artificial samples could be divided into two classes: artificial samples produced with the assistance of the registered user and artificial samples produced without his assistance.

(ii) *A latent sample reactivation* risk relates to touch fingerprint systems. The rarely used sweep sensors do not have to address this problem, because the method for imaging includes a self-cleaning function during each capture. In addition to the liveness detection, touch fingerprint systems should include a cleaning mechanism after each imaging. It is worth noting that a biometric system should reveal under no circumstances the information (like a score or threshold) to the user that could be useful for attackers. Additionally, before initiating another transaction, a biometric system should clear all biometric data from the memory, to ensure its security in a case when the attacker gains control of the system.

(iii) Last but not least, there is a *severed sample*—as mentioned before.

To overcome the above mentioned threats or risks, the *liveness detection* [24] should be implemented. In an environment where a higher level of security is required, a biometric system should be a part of two- or three-factor authentication solution. There are other attacks at the sensor level but they are possible whether or not liveness quality is implemented.

The liveness testing may take place at the acquisition stage, when nonlive data are not acquired, or at the processing stage, when nonlive data are not processed.

Another very important feature for the security and proper working of a biometric system is to assure that the capture of the biometric sample and measurement of liveness occur at the same point in space and time [11]. Otherwise, an attacker may present his live biometrics to pass the liveness testing and then he may deceive the verification process by supplying an artificial sample.

It is clear not only from [4, 9] that the production of a fake finger (print) is very simple. Our own experiments have shown that to acquire some images (e.g., from glass, CD, film, or even paper) is not very difficult and, in addition, such image could be enhanced and postprocessed, which leads to a high-quality fingerprint. The following production process of a fake finger (print) is simple and can be accomplished in several hours. After that, it is possible to claim the identity as an impostor user and common (nearly all) fingerprint recognition systems confirm this false identity supported by such fake finger. Therefore, the application of *liveness detection* methods is a very important task and should be implemented (not only) in all systems with higher security requirements, such as border passport control systems and bank systems.

2. New Optical Methods for Liveness Detection

In the following subsection, three new principles of liveness detection will be described, which are based on optical changes— the first is based on fine movements of fingertip area measured by a laser triangulation module (Section 2.1); the second one is based on measurement of optical characteristics after pressure change on a finger (Section 2.2); the last one is based on measurement of optical changes based on illumination of a finger by light with various wavelengths (Section 2.3).

2.1. Measurement of Pulse Based on Optical Measurement. The interaction of light with matter (here with human tissue) can be basically described in the terms of absorption, scattering, and fluorescence. The light from the illumination like LED, laser, or other sources hits the skin and is partially scattered on the surface and partially enters the tissue in which it will be absorbed, scattered, or reemitted. So when illuminating the finger from the side a fraction of scattered and reemitted light leaves the finger in approximately 4π direction and can be detected in other direction. Such scattered light may include information about dynamical processes like blood flow, hemoglobin saturation, and pulse from inside the finger. Scanners based on this technique try to detect whether the scanned object exhibits characteristics of the pulse and blood flow consistent with a live human being [11]. It is not very difficult to determine whether the object indicates some kind of pulse and blood flow, but it is very difficult to decide if the acquired characteristics are coincident with a live sample. As a result, it is difficult to create an acceptance range of the sensor, which would lead to small error rates. The main problem is that the pulse of a human user varies from person to person—it depends on the emotional state of the person and also on the physical activities performed before the scanning procedure. In addition, the pulse and blood flow of the attacker's finger may be detected and accepted when a wafer-thin artificial sample is used.

Depending on the composition of the original light spectrum, the illuminated object appears in colors which result mainly from the absorption spectrum of the material secondly and much less from the angle of scattering and partially from material capability for fluorescence itself.

One example of an optical skin property is the scattering on skin surface and the absorption in the tissue mentioned above—the light illuminating the surface is partly scattered and partly absorbed and reemitted. The light detector acquires the outgoing light spectrum which has been changed in intensity due to absorption and eventually due to the fluorescence of the proteins in the tissue. Another example for optical skin feature is the saturation of hemoglobin [14, 17, 25], which binds oxygen molecules. When blood comes from the heart, oxygen molecules are bound to the hemoglobin (oxyhemoglobin), and, vice versa, when blood is flowing back to the heart, it is less saturated by oxygen (deoxyhemoglobin). The absorption properties of these two hemoglobin molecules are different that means the color of oxygenated blood is different from that of nonoxygenated blood. If we use a light source to illuminate the finger tissue, we can follow the blood flow based on the detection of oxygenated and nonoxygenated blood, respectively [17].

Another solution is proposed in [17] based on the analysis of movements of papillary lines of the fingertips with the help of the so-called high precise laser distance measurement technique. One advantage of this implementation is that the finger is not required to be in contact with a specific measuring device, and so it can be integrated with standard fingerprint sensors, Figure 2.

The laser distance measurement [17, 26] module is placed to the right side of the glass plate, which is L-shaped. The user places his finger in such a way that it is in contact with the horizontal and the vertical side of the glass plate.

The underlying physical measurement principle is as follows. A semiconductor laser is used to produce a laser beam which illuminates the finger. Because of the scattering of the coherent laser wave fronts on the fingertip skin, a part of the laser spot is reflected back to sensor (position sensing device). This light interferes there and produces characteristic speckle pattern whose dynamic change is used for very precise triangulation measurement. Although the method is not suitable for resolving the papillary lines itself, it measures the small dynamic fluctuations of the papillary lines due to the heart beat and muscle tonus.

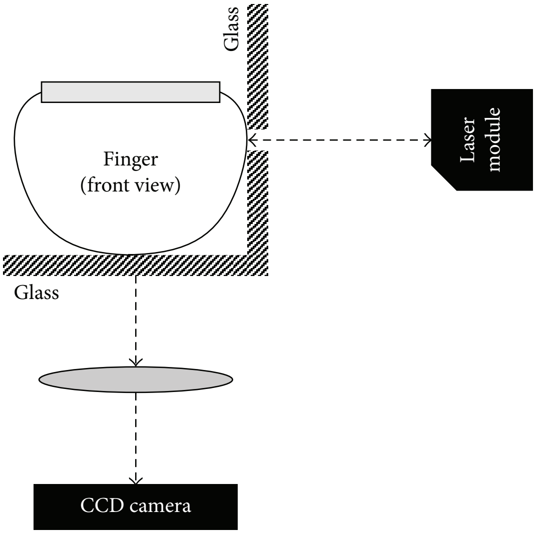


Figure 2: Possible integration of laser distance measurement for liveness detection with optical fingerprint sensor [17].

The comparison of the computed curve and a normalized standard curve (the template) will reveal whether the measurement corresponds to a standard live fingerprint or indicates a fake finger or another attempt of fraud. For example, the comparison between both curves can be realized by the normalization followed by the cross-correlation.

The optical bench consists from the following parts:

(i) Panasonic LM10 microlaser displacement sensor,

(ii) Panasonic LM10 sensor single comparator,

(iii) special holder for Panasonic LM10 sensor,

(iv) real-time digital phosphor oscilloscope Tektronix DPO7254.

A principle of the distance measurement using LM10 can be found on Figure 3.

The optical bench uses the Panasonic LM10 microlaser displacement sensor for measuring the distance between the finger and the sensor head. The measurement principle of LM10 is based on optical triangulation. A semiconductor laser is used to produce a laser beam which illuminates the finger. Because of the diffuse reflection of the laser beam on the fingertip skin, a part of laser rays is reflected back to sensor. A light spot from the reflected laser rays is tracked by a sensor part called position sensing device. The position sensing device measures the distance between fingerprint and sensor measuring the fluctuations of a light spot position. A diagram of the distance measurement process [27] can be seen in Figure 3.

The curve with heart activity (pulse) visible from the measurement using this method based on laser measurement principle can be seen in Figures 4 and 5.

In Figures 4 and 5 you can see the liveness detection results of two randomly chosen volunteers. Volunteer 1 is a woman, age 25. Volunteer 2 is a man, age 27. Both volunteers were before and during the experiments calm, rested, and in good physical and psychical condition. Results of this measurement are the curves with clearly visible heart activity (similar to classic electrocardiogram curve). As a matter of interest, it can be seen that the volunteer one (woman) has a faster pulse during the experiments.

There are other liveness detection methods based on optical principles—see the following sections.

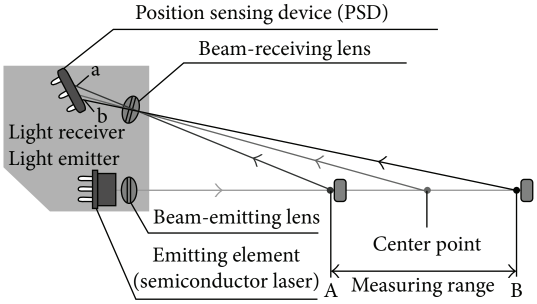


Figure 3: Measurement principle of the Panasonic LM10 microlaser displacement sensor [18].

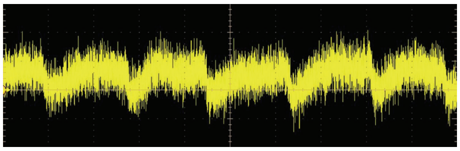


Figure 4: Result of liveness detection—volunteer 1.

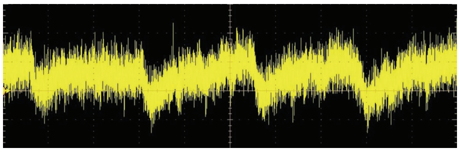


Figure 5: Result of liveness detection—volunteer 2.

2.2. Measurement of Optical Characteristics in the Finger Based on Pressure Change. In our team, a novel approach was proposed based on combination of detection of two characteristics of live human fingers (change of color and elasticity due to pressing of finger against glass plate)—closer information with testing results could be found in [28] and this section is based on [28].

Under normal circumstances, a live human finger is reddish and its papillary lines are approximately 0.2–0.5 mm wide (The width of papillary lines differs from one person to another, but it depends on various conditions, e.g., age of the person.). Due to the pressing of finger against glass plate, the height of papillary lines decreases so that the lines optically appear to be thicker and the blood is partly relocated from the pressed skin area so that the skin turns to yellowish/whitish [28, 29]. Once the pressure on the finger is decreased (or eliminated), the papillary line color and optical thickness immediately come closer (returns back) to its original state. However, the percentage of extension of width of papillary lines and its color are not always the same. The rate of change is proportional to the force of finger pressing.

The color of finger (and also the color change) can be detected using various color models. The experiments with various color models could be found in [28], for example, RGB, HLS, or CIE $L^* a^* b^*$. The results of experiments with HLS color model shows that this model is not convenient for purposes of this liveness detection due to the high intraclass variability.

The results of tests in case of CIE $L^* a^* b^*$ color model were much better. Due to pressing of finger against surface, the L^* value (lightness) is increased. The chromatic value a^*, which represents an axis from green to magenta, is significantly decreased and b^* chromatic value, which represents an axis from blue to yellow, is increased.

The results of RGB color model are more definite and proper. The biggest difference can be seen always between G components. The other differences are lower as follows [28]:

$$(G_2 - G_1) > (B_2 - B_1) > (R_2 - R_1), \qquad (1)$$

where x_2 is average value of X in center of image of pressed finger and x_1 is identical calculation for image of nonpressed finger, where X is particular component in RGB color model.

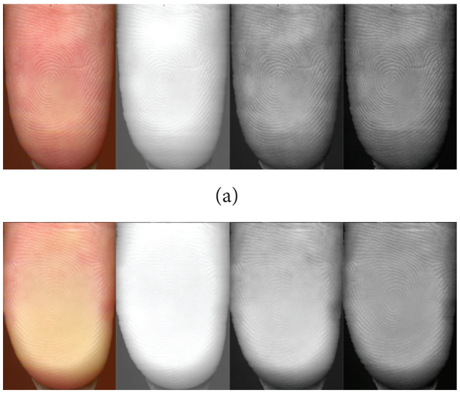
The optical comparison between nonpressed and pressed finger for full RGB image and also decomposed individual components can be found in Figure 6.

The width of papillary lines (and its change) could be detected in various ways. Above all, it will be necessary to choose an appropriate edge detector—a lot of edge detection methods are suitable for this purpose, for example, Sobel filter, Gabor filter, or Canny edge detector. The choice of appropriate method will be highly dependent on the used illumination source(s) mostly considering the angle of light. Moreover, the structure of used pipeline will be important, for example, use of appropriate image preprocessing/postprocessing techniques.

The successful liveness detection mechanism should meet a lot of requirements. According to the described biological principle of both tested characteristics of live human finger, the requirement for universality and permanence should be met. There is no expected problem according to the acceptability requirement. Nevertheless, in [28], these assumptions are checked during select tests by choosing of volunteers of different age, gender, and race and by tests of larger group of volunteers. The requirement for collectability was tested (and met). The requirement for concurrent measuring of the same area without interaction is met in the basis of the method proposal. The liveness detection measurements do not require any special illumination or other interfering hardware, so it is possible to run these measurements simultaneously with the capturing of fingerprint by common optical fingerprint sensor without any risk of negative interaction.

Regarding the requirement for security, it is necessary to ask for resistibility against the known methods of sensor spoofing. There is a lot of possible ways how to create an artificial finger of appropriate color, but there is no skin-color material, which will be able to change the color in the same way as the pressed finger.

The possible way how to pretend the color change is to exchange two fake fingers, each of a different color or to use two inks and to soak the stamp in the second ink during the capturing process. In [28], the exchange of two samples was tested using the Nikon camera (30 fps) and the speed of exchange was only 0.07 seconds. Nevertheless, this situation cannot spoof the proposed liveness detection unit, if the continuous monitoring of the color change will be implemented and the camera with high frame rate will be used.


(a)

(b)

Figure 6: Comparison of nonpressed finger (a) and pressed finger (b). In the first column (from the left), there is the finger in all RGB colors, in the second one, there is only the R-channel, the G-channel is in the third, and the B-channel is in the fourth column. The difference between average R values is 11, G 42 and B 20 [28].

Forgery of change of papillary lines width is also a nontrivial task. The elasticity of materials for fake finger creation has not been so widely tested. Thus, it is not possible to exclude the eventuality that some of the commonly used materials can have similar properties to the live human skin.

Generally, the common materials usable for fake finger creation to imitate the elasticity of the live human skin can be divided into three groups [28]: *pressure resistant materials* (e.g., sheet of rubber from a common office stamp), *ordinary materials* (e.g., gelatin or latex are often used), and *soft* (easily deformable) *materials*. In case of pressure resistant materials, the change of papillary line width should not be visible. It seems logical to use soft/easily deformable materials and to forge the change of papillary line width by controlling the pressing force. However, such fake fingers are often not able to forge the reverse change (decrease of the pressure and lifting of finger from the sensor surface) due to the slow or even nonexisting memory effect of material. Nevertheless, it is necessary to test various materials during the tests of this approach.

Another possible approach about how to imitate the change of width and color of papillary lines could be the use of thin semitransparent fake finger. Nevertheless, the creation and use of such fake finger could be very difficult (or even impossible), because there are two opposing requirements for the level of transparency. These fake fingers have to be transparent enough to allow to clearly see the color change, and nontransparent enough to allow to clearly see the papillary lines on the fake finger surface noninterfering with the papillary lines from the live finger behind [28].

Moreover, it is necessary to take into account that if the material is not as hard as glass, the finger has to be pressed significantly stronger to achieve the same color change, which influences the change of width of papillary lines on the fake finger surface. Another possible complication for the attacker could be the fact that a lot of commonly used transparent (or semitransparent) materials often contain significant amount of bubbles.

One of very often discussed ways how to spoof fingerprint sensor is the use of dead finger [28]. The capturing of the dead/removed finger may be difficult. It is also known that the color of human skin is conditioned by the circulation of the blood and that the skin due to the lack of blood circulation turns pale/grayish (pallor mortis). According [30], the paleness of skin develops rapidly and it can be easily optically distinguished from the common live skin color. The following postmortem change of skin color is turning dark purple (livor mortis) [30]. This change is caused by gravity and thus it is present only in the lower part of the body. During the first few hours after death, the dark purple parts of skin can turn whitish after applying pressure, but later, this effect is not observable.

There are many physical, chemical, and bacterial decay changes setting after death [27, 31, 32]. Color of the body is ultimately determined by the degree of oxyhemoglobin in the blood present at the time of death. With decreasing oxygen in the blood, the coloration of the skin is changing in the most cases from pink to pale and purplish blue—*pallor* and *livor mortis*. The following color changes are caused by degradation of hemoglobin. Decomposition of soft tissues is coming immediately. It is a process of endogenous autolysis (process of self-digestion by enzymes) and putrefaction (caused by bacterial flora owing to the fact that the body no longer has a functional immune system). Decomposition is progressing with breakdown of blood vessels and extravasation of red blood cells into the subcutaneous and adipose tissues. Soft tissues including skin are going to be disintegrated. Epidermal vesicle formation and skin slippage occur as the epidermis separates from the underlying dermis. Nevertheless, the epidermis commonly retains enough ridge detail to allow fingerprints to be obtained. It assists in the identification of the decedent but also could potentially make impossible identifying a genuine user of biometric systems.

According to the above described color changes of dead skin, this liveness detection approach could be capable to identify the dead finger as a fake finger. Generally speaking, the elasticity could be a little bit weaker than the color change, but coupled together they could create very strong barrier for the possible impostor. The proposed approach could also deal with the capturing of dry, wet, or bended skin, which can be an advantage in comparison with other approaches. Another advantage of this approach is that this method needs not wait until some physiological process (e.g., perspiration or several heartbeats) takes place. When using the hardware with appropriate parameters, the speed of the whole system is limited only by the quality of algorithm implementation. On the other hand, there is also a disadvantage. The proposed approach can have a problem with a high percentage of skin contaminated by colored material (e.g., ink, chalk, or some chemical substances), so the possibilities of deployment of this sensor could be slightly limited. On the other hand, a lot of sensors on the market have a similar problem.

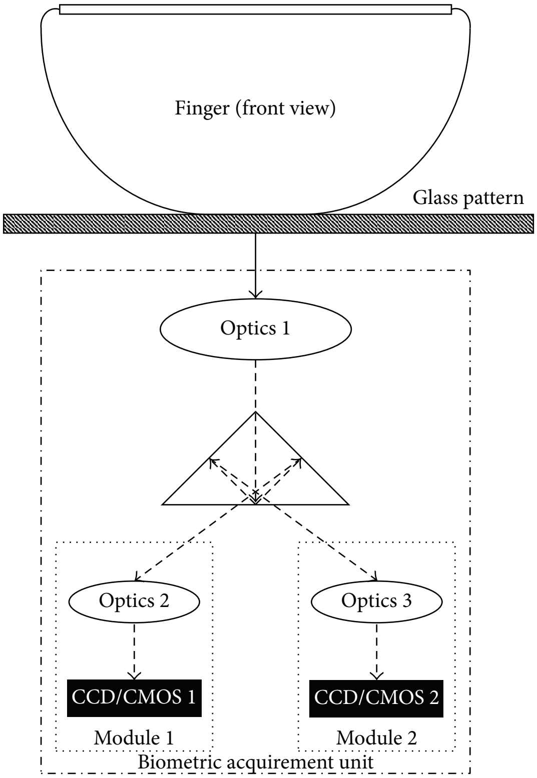


Figure 7: Schema of the proposed sensor.

According to the previously described requirements and the software principle of new method, we proposed the hardware schema of the possible liveness detection unit (This method was registered as the Czech utility model no. 19364 by the Czech Industrial Property Office in 2009.), see Figure 7. This unit can be integrated into an optical fingerprint sensor or it can be used as a sensor with the liveness detection ability (after a few necessary adjustments). In comparison with other partially similar approaches, the proposed liveness detection unit does not need any specific illumination sources, and the common white LED diodes or other ordinary light sources in various locations are sufficient.

The whole unit consists of two camera modules, prism, optics, and glass plate; see Figure 7. The first camera (camera module) will be used for detection of papillary lines width. It is necessary to use the camera with good quality optics to achieve the sufficient magnification of papillary lines, but the camera can have lower image framerate and it can use gray-scale image/video stream. The second camera has to follow the process of color change, so it has to produce a video stream with color images (lower resolution is possible). Nevertheless, this camera will be also used for detection of possible attacks (e.g., by exchanging two different artificial fingers). Because this kind of attack can be done quickly, it is necessary to have the camera with high image framerate (30 fps or better).

It is possible to use only one camera module, but in such case, the "united module" would have to meet all requirements for both separate camera modules. Such solution is currently more expensive and the possibilities of miniaturization are limited as well.

For the purposes of testing of this approach, a new optical bench was created. The bench consists of body, camera mounting module, camera (or other capturing device or other sensor generally), special fingerprint module, and mounting module. This optical bench is designed as multifunctional, so both mounting modules allow to set an arbitrary position (in the corresponding axis) and also to mount different sensors/fingerprint modules, so the whole unit can be used for testing of different configurations and even different ideas (not even for the liveness detection purposes).

A special fingerprint module was developed for the purposes of testing of this approach. This module is intentionally robust, because during preliminary tests, volunteers often feared that they could destroy the facility by pressing too hard. For higher user friendliness, the module has an entrance for a finger from both sides.

2.3. Measurement of Optical Changes in the Finger Based on Illumination with Various Wavelengths. A liveness detection method described in the Section 2.2 has a significant disadvantage—it requires a contact between finger and the fingerprint scanner glass. Therefore, its usage is practically limited to touch-based optical sensors. Other types of fingerprint scanners, for example, capacitive scanners, pressure based scanners, e-field scanners, and so forth, cannot use a contact based method because their surface containing capacitors, electro-conductive layers, or small electric field measuring antennas do not allow the presence of glass. Nowadays, also the touchless fingerprint scanners including those based on optical sensing technology are widely used. Contact-based liveness detection methods cannot be used for these sensors from obvious reasons.

Fortunately, a *contactless liveness detection method* compatible with all kinds of fingerprint sensors has been developed [33]. It is based on light illumination with various wavelengths and optical changes measurement. The source of light and the cameras do not need to be placed directly under the scanned finger where usually the fingerprint scanner is placed. For example, the source of light can be placed on one side of scanning part and the camera on the other side. This is the most convenient method for all touchless scanners.

This method is based on the following principle. A light of specific wavelength emitted from a light source mounted into fingerprint scanner illuminates a scanned fingertip. In this way, the part of emitted electromagnetic radiation encountering some object is absorbed by the material, part of this light is reflected back, and part of the light goes through the object. The main point of our interest is a light reflected from the fingertip surface and from the nearby sublayers. For the liveness detection purposes, an amount of the reflected part of light is measured. A live finger due to its physical attributes (blood oxygenation, temperature, etc.) has different spectral properties than the dead or fake finger. Also the amount of reflected light of a specific wavelength should differentiate the live finger from the other ones.

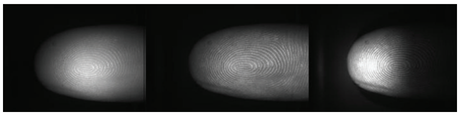


Figure 8: Finger illuminated by 700 nm red light, 550 nm green light, and 470 nm blue light.

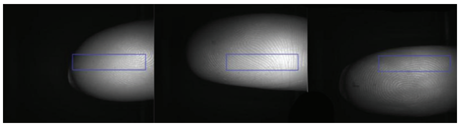


Figure 9: Rectangle for feature extraction.

For the experiments, the following light wavelengths were selected: 700 nm (red), 550 nm (green), and 470 nm (blue). The illuminated fingerprints were captured by a high-resolution camera and saved as a 1,280 × 1,024 pixel grayscale images. The example of scanned fingerprints can be seen in Figure 8.

Before determination of suitable features, an image preprocessing has to be performed. The fingerprint image background has to be excluded from further processing because the black background pixels can strongly influence the global features extracted from the image used for liveness detection. After scanning a large amount of fingerprints, we were able to determine a fixed region of interest (ROI) in the image invariant to the finger position where no background is present. The example of different finger positions with the determined ROI highlighted can be seen in Figure 9. This ROI is used afterwards for the feature detection. From the determined rectangle, it is possible to extract local and the global features.

As a local feature, for example, a pixel intensity arithmetic mean or the pixel intensity standard deviation from only some selected pixels could be considered. During our experiments, we determined three line segments inside the rectangle, each segment containing 11 pixels of interest from which the local features were extracted. The visualization of line segment position and pixels of interest position can be seen in Figure 10.

As a global feature extracted from the whole rectangle, the following statistics indicators were considered: pixel intensity arithmetic mean, pixel intensity standard deviation, pixel intensity median, histogram mean, histogram standard deviation, and the histogram median.

Having captured the three grayscale images of finger illuminated by three different wavelengths, it is possible to merge these three one-channel images into one three-channel color image. Some significant features could be extracted. We used the analysis of 3-channel histogram. During our experiments, we considered several color models. By simply merging the input images, we got the RGB representation of fingerprint. Image in RGB gained by merging the images from Figure 8 can be seen in Figure 11.

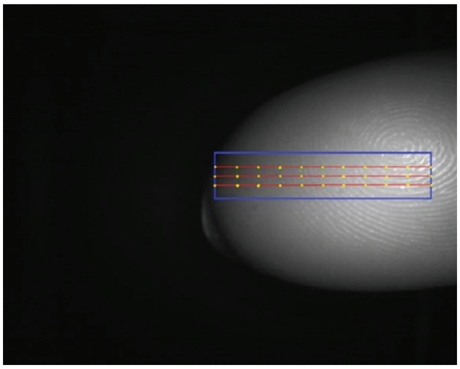


Figure 10: Local feature extraction points.

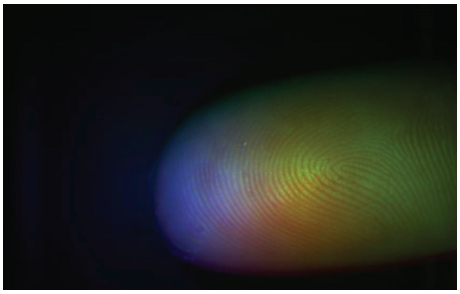


Figure 11: Color image obtained by merging the original images (RGB color model).

In image processing and computer vision areas, the RGB color model is not always the best option. By converting the color image into some other color models, the precision of live-finger decision could be significantly increased. The following color models were considered: XYZ color model (1931 CIE), LUV color model (CIE 1976), LAB color model (CIE 1976), and the YCbCr color model. Figure 12 shows the results of the color model conversions. The previously mentioned features can be extracted from any color image.

All the extracted features from a specific color model are concatenated to one feature vector. Feature vectors from various real and artificial fingers are used as the training data for the selected machine-learning method. For our purposes, we use artificial neural networks and random forests [13, 34].

We evaluated performance of each proposed algorithm by 8 cross-validation runs on the database containing 150 fingerprints [33]. The best result was achieved by combination of Luv and YCbCr model. The FRR was less than 2% while FAR was 10%. These results can be further improved by applying more LED with different wavelengths and more advanced feature extractors.

2.4. Skin Diseases and Their Influence on Liveness Detection.

Skin diseases could represent a serious problem in the process of liveness detection. In a general medicine, about 20%–25% of patients suffer from some skin disorder. In this paper, we discuss the diseases localized on palmar side of hands including fingertips. Some diagnoses are typical for this localization; others affect skin surface generally comprehending palms and fingers. When discussing whether the fingerprint recognition technology is a perfect solution capable to resolve all security problems, we should always keep in mind those potential users who suffer from some skin diseases.

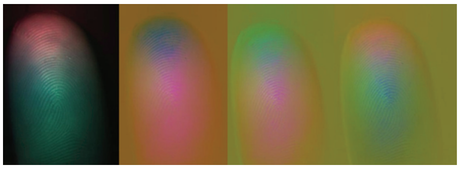


Figure 12: Color image converted into XYZ, LUV, LAB, and YCbCr color model.

The border of *epidermis* and *dermis* (dermoepidermal junction) forms the base of papillary lines. In most cases of dermatological disorders, we find a lot of changes in the ultrastructure of the skin, including epidermis and dermis. There is often inflammation, atrophy or hypertrophy, fibrotisation, and many other changes visible in the microscope. These differences result in changes of color (optical characteristics), changes of dermal vessels and capillaries (blood perfusion), and changes of elasticity and thickness of the skin (optical characteristics after pressure change).

Some examples of skin diseases potentially making impossible the liveness detection are the following ones. The description is made from a medical point of view.

Hand and fingertip eczema [35, 36] (see Figure 13(a)) is an inflammatory noninfectious long-lasting disease with relapsing course. It is one of the most common problems encountered by the dermatologist. Hand dermatitis causes discomfort and embarrassment and, because of its locations, interferes significantly with normal daily activities. Hand dermatitis is common in industrial occupations. The prevalence of hand eczema was approximately 5.4% and was twice as common in females as in males. The most common type of hand eczema was irritant contact dermatitis (35%), followed by atopic eczema (22%) and allergic contact dermatitis (19%). The most common contact allergies were to nickel, cobalt, fragrance mix, balsam of Peru, and colophony. Hand eczema was more common among people reporting occupational exposure. The most harmful exposure was to chemicals, water and detergents, dust, and dry dirt.

Warts (*verruca vulgaris*) [29, 37] (see Figure 14(a)) are benign epidermal neoplasms that are caused by human papilloma viruses (HPVs). Warts commonly appear at sites of trauma, on the hand, in periungual regions. HPVs induce hyperplasia and hyperkeratosis.

Psoriasis [27, 38] (see Figure 13(b)) is characterized by scaly papules and plaques. It occurs in 1% to 3% of the population. The disease is transmitted genetically; environmental factors are needed to precipitate the disease. Psoriasis of the palms and fingertips is characterized by red plaques with thick grey scale and may be indistinguishable from chronic eczema.

Pompholyx (*dyshidrosis*) [39] (see Figure 13(b)) is a distinctive reaction pattern of unknown etiology presenting as symmetric vesicular hand and foot dermatitis. Itching

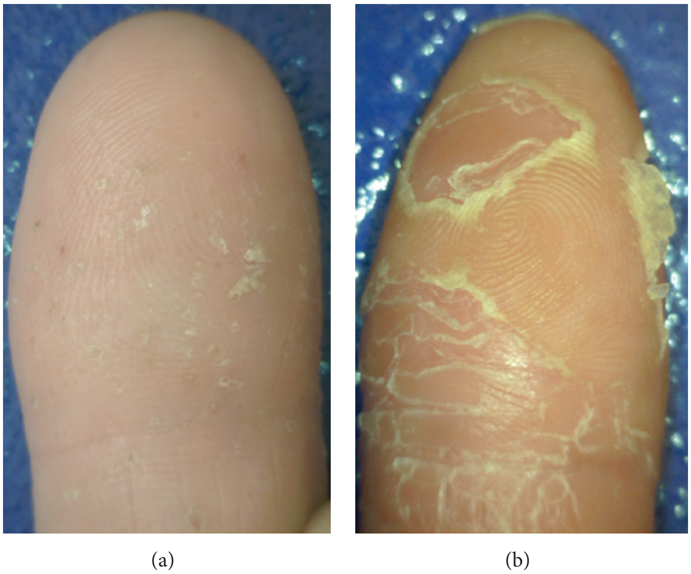


Figure 13: (a) Fingertip eczema; (b) psoriasis.

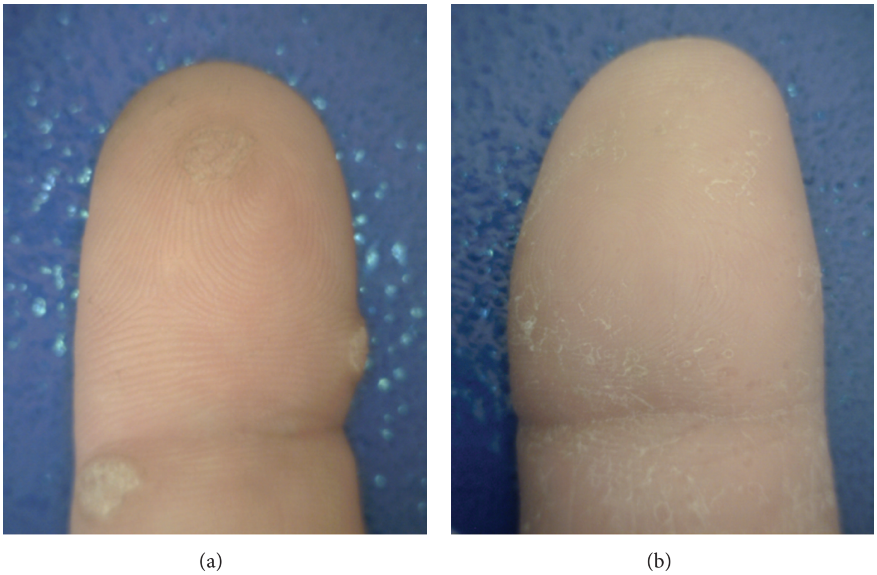


Figure 14: (a) Verruca vulgaris; (b) dyshidrosis.

precedes the appearance of vesicles on the palms and sides of the fingers. The skin may be red and wet.

Tinea of the palm [35] is dry, diffuse, keratotic form of tinea. The dry keratotic form may be asymptomatic and the patient may be unaware of the infection, attributing the dry, thick, scaly surface to hard physical labor. It is frequently seen in association with tinea pedis whose prevalence is 10 to 30%.

Pyoderma [29] is a sign of bacterial infection of the skin. It is caused by *Staphylococcus aureus* and *Streptococcus pyogenes*. Some people are more susceptible to these diseases (such as diabetics, alcoholics, etc.).

Systemic sclerosis [27] is a chronic autoimmune disease characterized by sclerosis of the skin or other organs. Emergence of acrosclerosis is decisive for fingerprinting. Initially the skin is infused with edema mainly affecting hands. With the progressive edema stiff skin appears and necrosis of fingers may form. For more than 90% of patients is typical Raynaud's phenomenon (see below). The typical patient is a woman over 50 years of age.

Raynaud's phenomenon [35] represents an episodic vasoconstriction of the digital arteries and arterioles that is precipitated by cold and stress. There are three stages during a single episode: pallor (white), cyanosis (blue), and hyperemia (red).

Erythema multiforme [29] is quite common skin disorder with multifactorial cause. The most common triggering agents are infections (in the first place herpes virus) and drugs. Both forms are characterized by erythematous target-shaped lesions with a center with hemorrhage, blistering, necrosis, or crust.

Epidermolysis bullosa [27] is a term given to groups of genetic diseases in which minor trauma causes noninflammatory blistering (mechanobullosus diseases). Repetitive trauma may lead to a mitten-like deformity with digits encased in an epidermal "cocoon." These diseases are classified as scarring and nonscarring and histologically by the level of blister formation.

Dermatitis artefacta [35, 40, 41] are changes of skin due to the manipulation by patient. Patients often have psychosomatic, psychiatric, or drug abuse problems.

Cutaneous adverse drug reactions [35] occur in many forms and can mimic virtually any dermatosis and they occur in 2%-3% of hospitalized patients. Antibiotics, sulfonamides, some nonsteroidal antiphlogistics, and anticonvulsants are most often applied in the etiology.

3. Conclusion

This paper introduces fake finger use and liveness detection in general at the beginning. The ways of possible attacks to fingerprint-based biometric systems are described as well. The main part of this paper is devoted to three new methods of liveness detection suitable for fingerprint recognition systems. The first method is based on pulse detection (heart activity), which is measured by a laser distance measurement unit (triangulation principle)—the pulse curvature in the acquired data is comparable with ECG signal. This first method is patented by us—see [26]. The second approach is based on detection of color change of the fingertip skin and thickness of papillary lines after the application of higher pressure to a glass platen. The color change could be distinguished in various color models, for example, RGB. The second method is registered by us as a national utility model by the Czech Industrial Property Office under the no. 19364 (http://isdv.upv.cz/portal/pls/portal/portlets.pts.det?xprim=1064464) (2009). The last method is based on reaction of skin on the fingertip to illumination with various wavelengths—this principle is not fully new, and the company Lumidigm Inc. has some patents in this area. Anyway we realized new experiments and these are published in this paper for the first time. At the end of this paper, the possible influence of skin diseases on liveness detection is discussed, because the skin diseases could really influence the liveness detection so that the live finger could be (due to any dermatologic problem) classified as a nonliving finger or artificial fake finger. The most influencing skin diseases are described in the last section.

Conflict of Interests

The authors declare that they have no conflict of interests.

Acknowledgments

This research has been realized under the support of the following grants: "*International Clinical Research Center*"—FNUSA ICRC CZ.1.05/1.1.00/02.0123, "*Security-Oriented Research in Information Technology*"—MSM0021630528 (CZ), "*The IT4-Innovations Centre of Excellence*"—IT4I-CZ 1.05/1.1.00/02.0070, and by Dongguk University Fund.

References

[1] Z. Říha and V. Matyáš, *Biometric Authentication Systems*, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2000.

[2] M. Sandström, *Liveness Detection in Fingerprint Recognition Systems [Diploma thesis]*, Linköping University, Linköping, Sweden, 2004.

[3] M. Drahanský, R. Nötzel, and K. W. Bonfig, *Sensoren zur Fingerabdruckerkennung*, bQuadrat, Germany, 2004.

[4] M. Drahanský, *Fingerprint recognition technology: liveness detection, image quality and skin diseases [Habilitation thesis]*, Brno, Czech Republic, 2010.

[5] P. D. Wasserman, *Solid-State Fingerprint Scanners*, Presentation, NIST, 2005.

[6] http://biolab.csr.unibo.it/.

[7] C. M. Oddo, L. Beccai, M. Felder, F. Giovacchini, and M. C. Carrozza, "Artificial roughness encoding with a bio-inspired MEMS-based tactile sensor array," *Sensors*, vol. 9, no. 5, pp. 3161–3183, 2009.

[8] M. Drahanský, "Liveness detection in biometrics," in *Advances Biometric Technologies*, 2011.

[9] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial "gummy" fingers on fingerprint systems," in *Optical Security and Counterfeit Deterrence Techniques IV*, Proceedings of SPIE, pp. 275–289, January 2002.

[10] C. Roberts, "Biometric attack vectors and defences," *Computers and Security*, vol. 26, no. 1, pp. 14–25, 2007.

[11] M. Kluz, *Liveness testing in biometric systems [M.S. thesis]*, Faculty of Informatics, Masaryk Universit, Brno, Czech Republic, 2005.

[12] R. Seshadri and Y. K. Avulapati, "Concealing the level-3 features of fingerprint in a facial image," *International Journal on Computer Science and Engineering*, vol. 2, no. 8, pp. 2742–2744, 2010.

[13] K. Zebbiche and F. Khelifi, "Region-based watermarking of biometric images: case study in fingerprint images," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, Article ID 492942, 13 pages, 2008.

[14] J. G. Pak and K. H. Park, "Advanced pulse oximetry system for remote monitoring and management," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 930582, 8 pages, 2012.

[15] Y. An, "Security analysis and enhancement of an effective biometric-based remote user authentication scheme using smart cards," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 519723, 6 pages, 2012.

[16] Fingerprint Verification Competition 2002, 2004, 2006, http://bias.csr.unibo.it/fvc2002/.

[17] M. Drahanský, W. Funk, and R. Nötzel, "Liveness detection based on fine movements of the fingertip surface," in *Proceedings of the IEEE West Point Workshop*, pp. 42–47, West Point, NY, USA, 2006.

[18] http://www.panasonic-electric-works.com/peweu/en/html/lm10.php.

[19] A. Chaudhari and P. J. Deore, "Spoof attack detection in fingerprint biometric system using histogram features," *World Journal of Science and Technology*, vol. 2, no. 4, 2012.

[20] M. Hariri and S. B. Shokouhi, "Robustness of multi-biometric authentication systems against spoofing," *Journal of Computer and Information Science*, vol. 5, no. 1, pp. 77–86, 2012.

[21] T. K. Neela and K. S. Kahlon, "A framework for authentication using fingerprint and electroencephalogram as biometrics modalities," *International Journal of Computer Science and Management Research*, vol. 1, no. 1, 2012.

[22] IBG, *Liveness Detection in Biometric Systems*, International Biometric Group, 2008, http://www.biometricgroup.com/.

[23] A. K. Jain, *Biometric System Security*, Michigan State University, 2005.

[24] B. G. Warwante and S. A. Maske, "Wavelet based fingerprint liveness detection," *International Journal of Engineering Research and Applications*, vol. 2, no. 2, pp. 1643–1645, 2012.

[25] R. C. Puffer and F. Kallmes, "Importance of continuous pulse oximetry of the ipsilateral thumb/index finger during transradial angiography," *Case Reports in Anesthesiology*, vol. 2011, Article ID 653625, 3 pages, 2011.

[26] M. Drahanský, W. Funk, and R. Nötzel, "Method and Apparatus for Detecting Biometric Features," International PCT Patent, pub. no. WO/2007/036370, no. PCT/EP2006/009533, 2007, http://www.wipo.int/pctdb/en/wo.jsp?wo=2007036370&IA=WO2007036370&DISPLAY=STATUS.

[27] http://forensicmd.files.wordpress.com/2009/12/early-postmortem-changes1.pdf.

[28] D. Lodrová, *Security of biometric systems [Dissertation thesis]*, FIT BUT, Brno, Czech Republic, 2013.

[29] B. Schneier, "Safe Personal Computing," Crypto-gram, Counterpane, Internet Security, 2001, http://www.schneier.com/crypto-gram-0105.html.

[30] A. T. Shäfer, "Colour measurements of pallor mortis," *International Journal of Legal Medicine*, vol. 113, no. 2, pp. 81–83, 1999.

[31] http://www.wikiskripta.eu/index.php/Posmrtn%C3%A9_zm%C4%9Bny.

[32] http://emedicine.medscape.com/article/1680032-overview.

[33] T. Malý, *Detection of finger liveness using illumination with different wavelengths [Diploma thesis]*, FIT BUT, Brno, Czech Republic, 2013.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science Business Media, New York, NY, USA, 2006.

[35] T. P. Habif, *Clinical Dermatology*, Mosby, 4th edition, 2004.

[36] K. Wolff, R. A. Johnson, and D. Suurmond, *Fitzpatrick's Color Atlas and Synopsis of Clinical Dermatology*, McGraw-Hill, 5th edition, 2005.

[37] J. Štork, L. Lacina, and O. Kodet, *Dermatovenerologie*, Galén, Prague, Czech Republic, 2008.

[38] N. Solak Tekin, I. O. Tekin, F. Barut, and E. Yilmaz Sipahi, "Accumulation of oxidized low-density lipoprotein in psoriatic skin and changes of plasma lipid levels in psoriatic patients," *Mediators of Inflammation*, vol. 2007, Article ID 78454, 5 pages, 2007.

[39] W. D. James, T. G. Berger, and D. M. Elston, *Andrew's Diseases of the Skin—Clinical Dermatology*, Saunders Elsevier, 10th edition, 2006.

[40] M. Y. Z. Lau, J. A. Burgess, R. Nixon, S. C. Dharmage, and M. C. Matheson, "A review of the impact of occupational contact dermatitis in quality of life," *Journal of Allergy*, vol. 2011, Article ID 964509, 12 pages, 2011.

[41] M. S. Jeong, S. E. Choi, J. Y. Kim et al., "Atopic dermatitis-like skin lesions reduced by topical application and intraperitoneal injection of hirsutenone in NC/Nga mice," *Clinical and Developmental Immunology*, vol. 2010, Article ID 618517, 7 pages, 2010.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 427542, 7 pages
http://dx.doi.org/10.1155/2013/427542

Research Article

A Study on User Authentication Methodology Using Numeric Password and Fingerprint Biometric Information

Seung-hwan Ju,[1] Hee-suk Seo,[2] Sung-hyu Han,[3] Jae-cheol Ryou,[4] and Jin Kwak[5]

[1] *Research Institute, Korea Electric Power Corporation, Yuseong-Gu, Daejeon 305-706, Republic of Korea*
[2] *Department of Computer Engineering, Korea University of Technology and Education, Cheonan, Chungnam 330-708, Republic of Korea*
[3] *The Faculty of Liberal Arts, Korea University of Technology and Education, Cheonan, Chungnam 330-708, Republic of Korea*
[4] *Department of Computer Engineering, Chungnam National University, Yuseong-Gu, Daejeon 305-764, Republic of Korea*
[5] *Information Security Engineering, Soonchunhyang University, Asan, Chungnam 336-745, Republic of Korea*

Correspondence should be addressed to Hee-suk Seo; histone@kut.ac.kr

Received 5 July 2013; Accepted 11 August 2013

Academic Editor: Sabah Mohammed

The prevalence of computers and the development of the Internet made us able to easily access information. As people are concerned about user information security, the interest of the user authentication method is growing. The most common computer authentication method is the use of alphanumerical usernames and passwords. The password authentication systems currently used are easy, but only if you know the password, as the user authentication is vulnerable. User authentication using fingerprints, only the user with the information that is specific to the authentication security is strong. But there are disadvantage such as the user cannot change the authentication key. In this study, we proposed authentication methodology that combines numeric-based password and biometric-based fingerprint authentication system. Use the information in the user's fingerprint, authentication keys to obtain security. Also, using numeric-based password can to easily change the password; the authentication keys were designed to provide flexibility.

1. Introduction

User authentication is a procedure to check the validity of the identification presented by a user. This is a matter that a machine authenticates a person. This user authentication is usually composed of three types [1].

The first type of authentication method is the user knowledge-based authentication method; it is a way of authenticating with information that a user remembers such as password and, it is the most widely used method because it is easy to implement. The second type of authentication method is the user's own-based authentication with smart cards or access cards belonging to it. The third type of authentication method is authentication using a user's physical characteristics, and fingerprints and recognition are the representative examples.

A fingerprint refers to patterns formed with lines of uplifted pores that exist on the human palm. Lines uplifted as shown in Figure 1 are called ridges and the places caved between two ridges are valleys. A fingerprint is a series of ridges and valleys appearing at the end of a finger. The fingerprint recognition is a process of finding fingerprints showing the same flow by analyzing the flow of these ridges.

Biometric information required for recognition is called feature, and features appearing in fingerprints are especially called minutia. The minutia [2] is divided into two types of ending and bifurcation. Ending refers to the place where the flow of ridges is cut, and bifurcation is the place where two ridges become one ridge. One fingerprint image has more than one ending and bifurcation.

A user is authenticated by using the place of this ending and bifurcation. Compared to the other user authentication methods, this user authentication method using users' physical characteristics has strong security [3].

However, the user authentication method using users' physical characteristics has critical security vulnerability such as the user authentication key cannot be changed. The fingerprint, or iris, and so forth are said to be users' own information but the user authentication key must be able to be

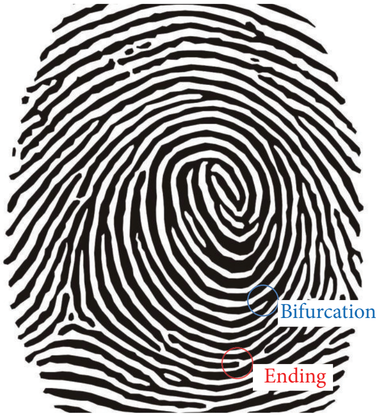


Figure 1: Elements of fingerprint—elements of the fingerprint to authenticate the user.

changed because when leaked, all the authentication systems using all of their biometric information are hopeless. As shown above, there are disadvantages to the user authentication method using biometric information such as fingerprint recognition, and therefore, this study tries to design biometric information so that it can be changed.

2. Study on Fingerprint Recognition User Authentication Mechanism

As shown in Figure 2, the fingerprint recognition mechanism goes through two steps of feature extraction and fingerprint matching.

The feature extraction step is the step for configuring minutiae data files to be used in the fingerprint matching step and is conducted in three steps of preprocessing, minutiae extraction, postprocessing, as shown in Figure 3.

2.1. Preprocessing

2.1.1. Image Improvement. A fingerprint image is classified into one of the images with a lot of noises. A fingerprint is a body part going through a lot of state changes such as injuries or moisture. Thus, fingerprint images obtained through the device are likely to be mixed with noises.

In the image improvement step, the work clarifying the distinction between ridges and valleys is carried out by reducing noises [4]. The most commonly used method is to use adaptive filter. It uses the fact that if knowing ridge local orientation around applied pixels and applying adaptive filter, ridges with the same direction become clear. In this process, the bridge of neighboring rides resulting from noises is removed and the result of connecting broken ridges is often shown. Directional Fourier filter [5], Gabor filter [6] and so forth, are widely used adaptive filters, and the method using mask operation is also used.

2.1.2. Binarization. When image improvement work is finished, the process of extracting ridges is started. As shown in Figure 4, fingerprint images usually have grayscale of 256 but this can be simplified into the binary information of ridges and valleys as binarized image of Figure 4(b).

There is a difficulty that binarization cannot be done by using single intensity threshold because all fingerprint images do not have constant image contrast in the process of making binary images, and even the contrast ratio of the same person's fingerprints varies every time the device is pressed on. Therefore, the dynamic thresholding method [7] is applied depending on image distribution pixel values and through it, the whole image is binarized into the ridge part and nonridge part.

2.1.3. Thinning. The final step of preprocessing to extract minutiae is the thinning step and this refers to the work reducing the width of ridges obtained after binarization into one pixel like minutiae extraction after thinning of Figure 4(d). This process must not only fully maintain coconnectivity of found ridges but minimize wrong minutiae information that may occur through this step. As can be seen in Figure 4(c) smoothened image, the flow of ridges becomes often clear by applying the smoothing technique to binary images. Many algorithms have been using this method because minutiae can be found quickly and easily through simple mask operations with thinned fingerprint images.

Preprocessing is the relatively time-consuming process. Since time consumption in the process of using adaptive filters and thinning accounts for the largest part, research on the algorithm which can ensure a high recognition rate while reducing the operation time of these two steps is needed.

2.2. Feature Extraction. After preprocessing is finished, the process of finding minutiae is carried out. As shown in Figure 5, by using thinned images, minutiae is distinguished by finding a point where a thin line ends for an endpoint and the point where three thin lines meet for bifurcation.

2.3. Postprocessing. The false minutia caused by the damage of the original image is included in the found minutiae and these are called false minutiae. Most false minutiae are created by incorrectly thinning the part where ridges are broken due to injuries and so forth, or the part where the shape of ridges is not shown well due to changes in binding force. By defining and removing false minutiae, postprocessing plays a role of reducing unnecessary operations in matching and increasing overall performance.

3. User Authentication of Number-Fingerprint Mapping

Recently, due to the rapid growth of the Internet with the development of computers, the need for personal authentication system at the private level which is easy to use while providing reliable security level has increased. Thus, developers came to develop algorithms and systems by focusing on the private demand of personal authentication, and many biometric authentication systems are currently commercialized and used. However, unlike other authentication methods, these biometric authentication systems have the disadvantage that they cannot be changed (keys or passwords are easy to change). Confidential authentication should be possible to

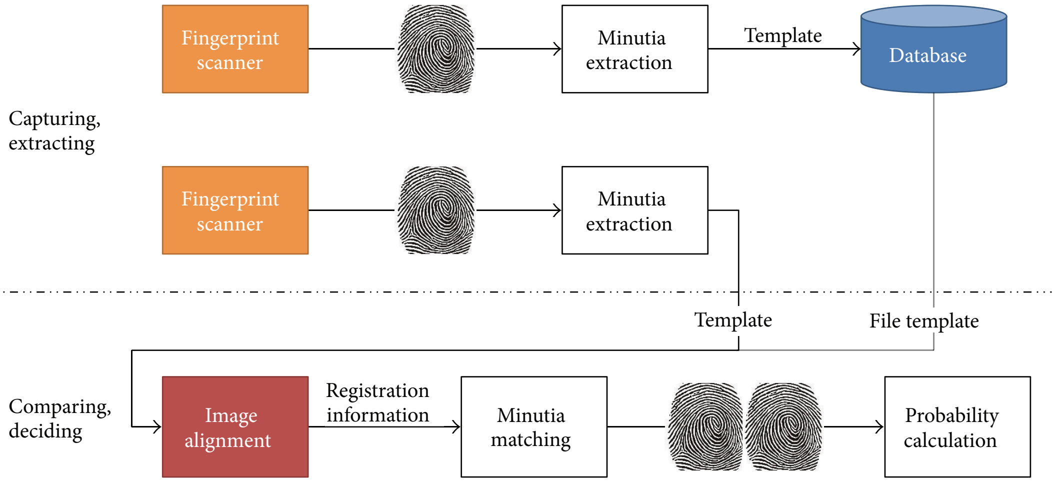


Figure 2: User authentication mechanism using fingerprint.

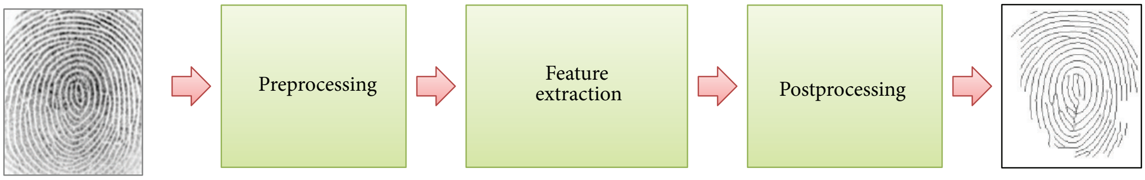
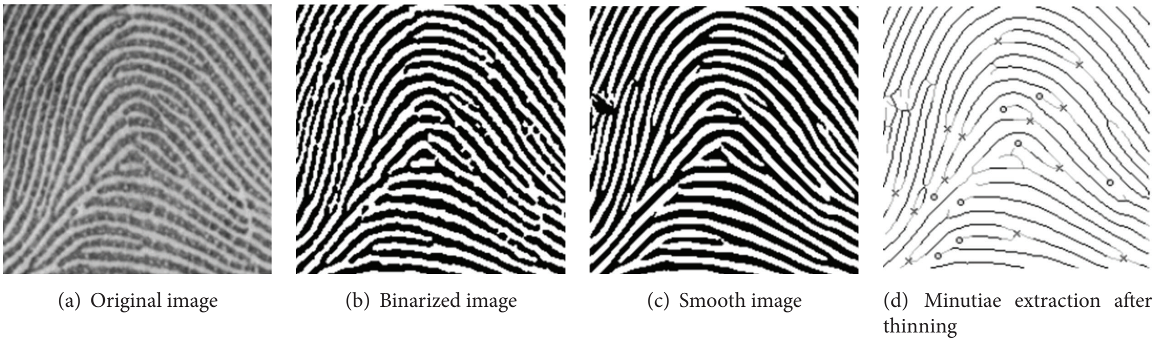


Figure 3: Procedures to fingerprint recognition.


(a) Original image

(b) Binarized image

(c) Smooth image

(d) Minutiae extraction after thinning

Figure 4: Preprocessing of fingerprint recognition.

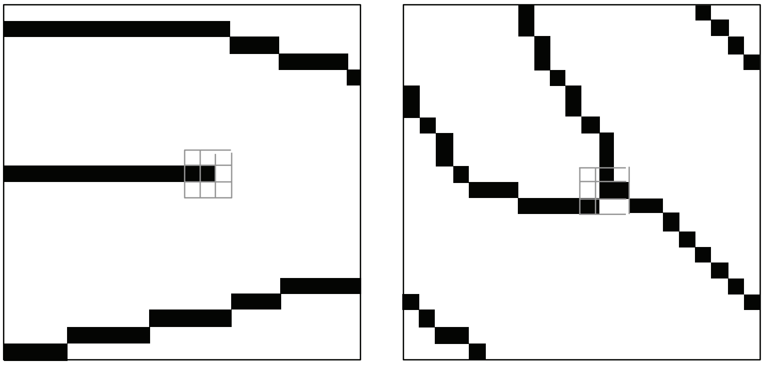


Figure 5: Feature extraction of fingerprint recognition.

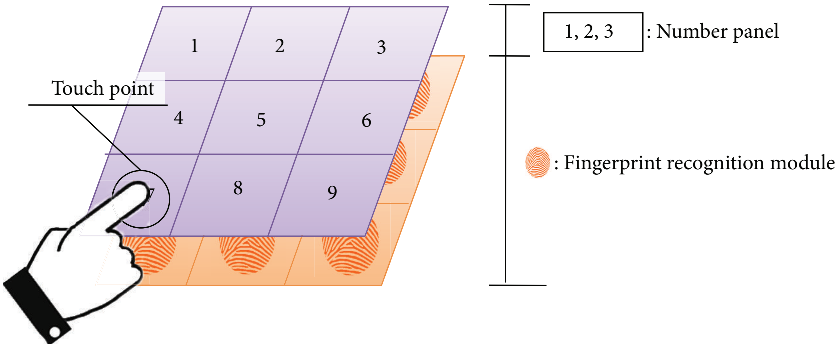


Figure 6: Schema of number-fingerprint authentication system.

change. In addition to the personal information leakage problem caused by biometric information leak, the biometric authentication technique such as fingerprint recognition cannot be changed. When their fingerprint information was leaked, all information recognized by computers can be copied and used. All the secrets entered by their fingerprint information come to nothing. Fingerprint information is no longer available, and it is highly likely to be abused. Therefore, their authentication information should be possible to change.

This study tries to propose the authentication system that can be changed by using number-based password and fingerprint biometric authentication.

3.1. Limitation of Fingerprint Recognition. In Table 1, 7 billion people of total world population are set as set P and 900,000 kinds of fingerprint reader results that can have 450,000 pixels and minutiae as $hash(P)$.

This operation can be regarded as a hash function because the result of operations is less than the total number of the population. The quotient of set Pdivided by $hash(P)$ is approximately 7777; therefore, 7777 people of the world's population may have the same fingerprint reader results. So, if including the entire world population, the existing fingerprint recognition system is vulnerable to security.

Therefore, the user authentication technique using bio-based biometric key has less risk of misuse because information itself has a close relationship with the owner along with the advantages of the existing auxiliary device. It is easy to use and hardly costs for maintenance and supplements the weakness of keys or identification tags because there is no risk of losing because it is always carried by the owner. However, in addition to the problem that it has the security vulnerabilities of hash function as they are, the bio-based user authentication technique has a security vulnerability such that a user authentication key cannot be changed.

The system applying fingerprint-based number password user authentication system presented in this paper has the following security strength.

If using a four-digit password, it has $900{,}000 \times 10^4$, that is, 9 billion number of cases and if using a six-digit password, $900{,}000 \times 10^6$, that is, 900 billion number of cases so if the entire world population of 7 billion people becomes users, sufficient security stability can be provided.

Table 1: An indicator of the limitations of fingerprint.

Index	Number
World population [8]	7 billion people
In 1.5 cm × 3.0 cm finger print, Fingerprint recognition at the interval of 0.001 mm	450,000 pixel
Minutiae found from fingerprints	Two kinds (Ending, bifurcation)

By having the advantages of both biometric-based user authentication technique and password-based user authentication technique, fingerprint recognition-based number password user authentication system can achieve both security and flexibility.

3.2. Number-Fingerprint Authentication System. By attaching the number panel on fingerprint recognition device, the user authentication system that uses number password and user fingerprint as authentication keys recognizes even the fingerprint of the user when a user is entering a password, see Figure 6.

By authenticating by mapping the user's fingerprint and number password in the user authentication system, we try to provide both flexibility of number password and security of biometric authentication.

4. Number-Fingerprint Mapping Digital Signature and Authentication

The applications of this system complexly applying fingerprint recognition and number panel are very diverse. The examples may be access control and attendance maintenance, PC security, e-commerce, and so forth. To be used in e-commerce, there should be algorithms on digital signatures and authentication. This paper will examine existing digital signatures and authentication methods and propose the algorithm that modified them for the system presented in this paper.

Digital signatures and corresponding authentication methods have been proposed very diversely [9]. Among them, the representative ones are digital signature using RSA, ElGamal method [10], Ong-Schnorr-Shamir method, signature and authentication method by ID of Shamir [11], and so forth. This paper will present RSA signature method and the ElGamal signature method by modifying them for the system presented in this paper.

4.1. Modified RSA Signature Method. First, after discussing the existing RSA signature method, this paper describes its modification. Existing RSA signature method can be summarized into the following three steps.

Step 1. The authentication center selects two large prime numbers p, q and calculates its multiplication $n = pq$. p, q are the values that only the authentication center knows and n is open to the public. The sender's private key e and public key p, q are calculated. e is the value that only sender and authentication center know and d is open to the public.

e and d must satisfy the following equation:

$$n^1 ed \equiv 1(\mathrm{mod} \emptyset (n)). \tag{1}$$

Here, $\emptyset(n) = (p-1)(q-1)$.

Step 2. The sender calculates the following for M, the message he/she wants to send.

Consider

$$S \equiv M^d \,(\mathrm{mod} n). \tag{2}$$

And, the sender sends M and S.

Step 3. The receiver receives M and S and then calculates the following:

$$M' \equiv S^e \,(\mathrm{mod} n). \tag{3}$$

If M and M' are the same, it is determined that there is no problem in authentication of M but if not it is determined that there is problem.

M and M' must be the same for the following reason:

$$S^e \equiv M^{de} \equiv M \,(\mathrm{mod} n). \tag{4}$$

The safety of the RSA signature method is based on the safety of the RSA public key cryptosystem. That is, it is based on the fact that when knowing n, the multiplication of prime numbers p, q, it is very difficult to factorize n.

The RSA signature method described above can be modified to be applied to the method described in this paper. There are two kinds of authentication processes in the methods that we described. One is fingerprint password and the other is number password. Let us say that fingerprint password is PW_1 and number password is PW_2. Now, modified RSA signature method can be described as follows.

Step 1. The authentication center selects two large prime numbers p, q and calculates its multiplication $n = pq$. p, q are the values that only the authentication center knows and n is open to the public. PW_1 is called d_1 and $PW_2 d_2$.

That is,

$$d_1 = PW_1, \qquad d_2 = PW_2. \tag{5}$$

$e_1$and e_2 are calculated to satisfy the following equation:

$$e_1 d_1 \equiv 1 \,(\mathrm{mod} \emptyset \,(n)), \qquad e_2 d_2 \equiv 1 \,(\mathrm{mod} \emptyset \,(n)). \tag{6}$$

And e_1 and e_2 are open to the public.

Step 2. The sender calculates the following for M, the message he/she wants to send.

Consider

$$\begin{aligned} d_1 S_1 &\equiv M^{d_1} \,(\mathrm{mod} n), \\ d_2 S_2 &\equiv M^{d_2} \,(\mathrm{mod} n). \end{aligned} \tag{7}$$

And, the sender sends M, S_1, and S_2.

Step 3. The receiver receives M, S_1, and S_2 and then calculates the following:

$$M_1 \equiv S_1^{e_1} \,(\mathrm{mod} n), \qquad M_2 \equiv S_2^{e_2} \,(\mathrm{mod} n). \tag{8}$$

And, the receiver checks if the following equation is established:

$$M = M_1, \qquad M = M_2. \tag{9}$$

Therefore, the following four cases may occur. They can be determined in several ways according to the policy of authentication system. The following shows one example:

(1) $M = M_1$, $M = M_2$: in this case, there is no problem because both fingerprints and number password are accurate.

(2) $M \neq M_1$, $M = M_2$: in this case, fingerprints are not accurate but number password is accurate. Therefore, if there is often an error in the fingerprint recognition system, authentication may be acceptable.

(3) $M = M_1$, $M \neq M_2$: in this case, fingerprints are accurate but number password is not accurate. Therefore, authentication may be accepted thinking that the sender entered wrong password by mistake.

(4) $M \neq M_1$, $M \neq M_2$: in this case, both are not correct. Therefore, it is determined that there is an error in authentication.

Of the four cases described above, the second case can be said to be very useful because an error often occurs in the fingerprint recognition system.

4.2. Modified ElGamal Signature Method. First, after discussing the existing ElGamal signature method, this paper describes its modification. Existing ElGamal signature method [2] can be summarized into the following three steps.

Step 1. The authentication center selects one large prime number P. Of $\{1, 2, 3, \ldots, p-1\}$, it selects primitive root g and then, it selects the sender's private key x and calculates the following:

$$y \equiv g^x \,(\mathrm{mod} p). \tag{10}$$

And g, y, and p are disclosed.

Step 2. The sender calculates the following for M, the message he/she wants to send. The sender selects arbitrary k, relative prime with $p-1$ and then calculates S and T as the values that satisfy the following equation:

$$\begin{gathered} S \equiv g^{k} (\bmod P), \\ M \equiv xS + kT (\bmod (P-1)). \end{gathered} \tag{11}$$

In (11), T can be calculated by using the Euclidean algorithm. T can be calculated because k was set as the relative prime with $p-1$ and M, S, and T are sent.

Step 3. The receiver receives M, S, and T and checks if the following equation is established:

$$g^{M} \equiv y^{S} S^{T} (\bmod p). \tag{12}$$

Equation (12) is established for the following reason:

$$g^{M} \equiv g^{xS+kT} \equiv g^{(xS)} g^{kT} \equiv y^{S} S^{T} (\bmod p). \tag{13}$$

When (13) is established, document signer authenticates it as legal.

While RSA method is based on the fact that prime factorization for large integers is difficult, ElGamal signature method is based on the fact that solving discrete logarithm problem in large prime number is difficult. Discrete logarithm problem means that even if calculating $y \equiv g^{x}$ for law p is easy when g and x are given, finding x satisfying $y \equiv g^{x}(\bmod p)$ is difficult when you know y and x. Like the signature method of modified RSA described above, the modified ElGamal signature method is described as follows. Let us suppose that fingerprint password is PW_1, and digit password is PW_2.

Step 1. The authentication center selects one large prime number p. Of $\{1, 2, 3, \ldots, p-1\}$, it selects primitive root g. PW_1 is called x_1 and $PW_2 x_2$. That is,

$$x_1 = PW_1, \qquad x_2 = PW_2. \tag{14}$$

And the following is calculated:

$$y_1 \equiv g^{x_1} (\bmod p), \qquad y_2 \equiv g^{x_2} (\bmod p). \tag{15}$$

And, g, y_1, y_2, and p are disclosed.

Step 2. The sender calculates the following for M, the message he/she wants to send. The sender selects arbitrary k, relative prime with $p-1$ and then calculates S and T_1, T_2 as the values that satisfy the following equation:

$$\begin{gathered} S \equiv g^{k} (\bmod p), \\ M \equiv xS + kT_1 (\bmod (p-1)), \\ M \equiv xS + kT_2 (\bmod (p-1)). \end{gathered} \tag{16}$$

And, M, S, T_1, and T_2 are sent.

Step 3. The receiver receives M, S, T_1, and T_2 and checks if the following two equations are established:

$$\begin{gathered} g^{M} \equiv y_1^{S} S^{T_1} (\bmod p), \\ g^{M} \equiv y_2^{S} S^{T_2} (\bmod p). \end{gathered} \tag{17}$$

Four cases similar to modified RSA signature method occur and each case can be determined in different ways according to the policy of the authentication system.

Until now, RSA signature method and ElGamal signature method have been examined, and also two signature methods were modified and applied to the authentication method of this paper. The advantage of the modified method is as follows. By using two ways of signature of fingerprint recognition and number password, errors caused by fingerprint recognition can be compensated.

5. Conclusion

With the rapid growth of computer and communication technology, users have access to information easier. Easier access to information, but the threat of information leakage has become increase. The personal authentication system is required, while providing information about the safety and security.

In this study, user authentication was performed that use biometric information and passwords of users. The user cannot change user's fingerprint information, but the user has a set password to easily change. So this authentication system provides security and flexibility.

Because it can make a password key that utilize the user's fingerprint and numeric password, an attacker does not have the advantage of leaked password.

In addition, it can remove authentication errors that recognize fingerprints among different users for the feature extraction results of two different users.

Acknowledgment

This research was funded by the Ministry of Science, ICT & Future Planning (MSIP), Korea, in the ICT R&D program 2013.

References

[1] S. H. Ju and H. S. Seo, "Password based user authentication methodology using multi-input on multi-touch environment," *Journal of the Korea Society For Simulation*, vol. 20, no. 1, pp. 39–49, 2011.

[2] G. I. Davida, Y. Frankel, and B. J. Matt, "On enabling secure applications through off-line biometric identification," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 148–157, May 1998.

[3] C. Lin and Y. Lai, "A flexible biometrics remote user authentication scheme," *Computer Standards and Interfaces*, vol. 27, no. 1, pp. 19–23, 2004.

[4] M. U. Akram, A. Tariq, S. A. Khan, and S. Nasir, "Fingerprint image: pre- and post-processing," *International Journal of Biometrics*, vol. 1, no. 1, pp. 63–80, 2008.

[5] B. G. Sherlock, D. M. Monro, and K. Millard, "Fingerprint enhancement by directional Fourier filtering," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 141, no. 2, pp. 87–94, 1994.

[6] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Gabor filter design for multiple texture segmentation," *Optical Engineering*, vol. 35, no. 10, pp. 2852–2863, 1996.

[7] A. M. Khan, W. Umar, T. Choudhary, F. Hussain, and M. H. Yousaf, "A new algorithmic approach for fingers detection and identification," in *International Conference on Graphic and Image Processing (ICGIP '12)*, vol. 8768 of *Proceedings of SPIE*, Singapore, March 2013.

[8] U.S. Census Bureau, "World POPClock Projection," 2013.

[9] S. G. Aki, "Digital signatures: a tutorial survey," *IEEE Computer Magazine*, vol. 16, no. 2, pp. 15–24, 1993.

[10] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[11] A. Shamir, "Identity-based cryptosystem and signature scheme," in *Proceeding of Crypto '84*, pp. 47–53, 1984.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 170580, 8 pages
http://dx.doi.org/10.1155/2013/170580

Research Article

Secure Encapsulation and Publication of Biological Services in the Cloud Computing Environment

Weizhe Zhang,[1] Xuehui Wang,[1] Bo Lu,[2] and Tai-hoon Kim[3]

[1] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[2] *Network and Information Center, Harbin Institute of Technology, Harbin 150001, China*
[3] *School of Computer and Information Science, University of Tasmania, Virginia Court, Sandy Bay, Hobart, TAS 7001, Australia*

Correspondence should be addressed to Weizhe Zhang; wzzhang@hit.edu.cn

Received 5 June 2013; Accepted 19 June 2013

Academic Editor: Sabah Mohammed

Secure encapsulation and publication for bioinformatics software products based on web service are presented, and the basic function of biological information is realized in the cloud computing environment. In the encapsulation phase, the workflow and function of bioinformatics software are conducted, the encapsulation interfaces are designed, and the runtime interaction between users and computers is simulated. In the publication phase, the execution and management mechanisms and principles of the GRAM components are analyzed. The functions such as remote user job submission and job status query are implemented by using the GRAM components. The services of bioinformatics software are published to remote users. Finally the basic prototype system of the biological cloud is achieved.

1. Introduction

In recent decade years, bioinformatics is a leading branch of biological science which deals with the study of methods for storing, retrieving, and analyzing biological data [1]. The rising of bioinformatics software becomes a specialized field. Some bioinformatics software products, such as Blast (database search) [2], Clustalw (multiple sequence match) [3], *phylip* (biological phylogenetic analysis software) [4], and EMBOSS (large sequence analysis) [5], have become an indispensable tool for molecular biology research. Open, public, and common are the main features of bioinformatics software. These software products are usually used in various forms of open source code copyright statement and take the Linux operating system as the main platform. They are mainly developed by scientific research institutions, colleges, universities, and other academic departments freely and provide public use. In the bioinformatics software's directory of GNU Project, there exist 15 kinds of software products, and, in the bioinformatics software's directory of the Open Science project, there exists 51 kinds.

However, the development of the open source software usually aims at solving some specific issues for a particular research field. The academic research is its main purpose, and personal interest is its main driving force. So the program often lacks detail documents and necessary user supports. Installation and configuration of this software will need a certain knowledge and experience of computer programming and system maintenance, which makes this software difficult to be used by most biologists in their research. Even if the user is biological information scientist or even the professional system administrator, it is difficult for them to face many software products with limited documents. In addition, with the rapid increasing of biological information's data scale and the further research for the data, some software will be needed in a research project [6]. The common bioinformatics software, including both free software from academic unit and expensive commercial software, far cannot satisfy the above requirements. For the data format, the same DNA or protein sequences in different databases have different storage format and use different input/output formats in different applications. The user must first be familiar with the conversion between these formats [7]. While, there are hundreds of kinds of sequence analysis software products, if the user is not familiar with their application and use methods, he needs to learn how to use the software and how to analyze the results, which is often wrong as well as time consuming. Although there have been many web-based analytical tools [8], the development of bioinformatics software still cannot get rid of the basic layout which takes the separate calculation method as basis, the individual computer program as the center, and the single calculation results as the goal, and some software's output results are difficult to understand for the biologists.

With the development of the web service in the cloud computing, we gradually pay more attention on this technology which contains a huge processing power. However, for the various bioinformatics products, even if we improve the processing power with the web service technology, we cannot solve the problem that the users need to spend a lot of time and energy to be familiar with different bioinformatics software's use methods or data organization structure. Therefore, it is necessary to provide unified and single software encapsulation for the bioinformatics software. Through the encapsulation, a unified interface and simple operation will be provided; thus the backstage software implementation details can be shielded off, and the users' requirements can be completed correctly and efficiently.

The following sections of this paper are organized as follows: Section 2 introduces the bioinformatics software objects to encapsulation and publication; Section 3 puts forward the outline of the bioinformatics software's encapsulation and publication and implements the encapsulation and publication on both Windows and Linux operating systems; Section 4 conducts functional tests for the bioinformatics software to check the correctness of the encapsulation and publication.

2. Example of Bioinformatics Software

This paper chooses three bioinformatics software products as example, namely, gene sequences conversion tools *seqret*, gene sequences ORF search tools *getorf*, and the molecular clock based maximum likelihood estimation tools *proml*. Among them, the two executive programs *seqret* and *getorf* are, respectively, encapsulated on both Windows and Linux operating systems. *Proml* is an application running on the Windows platform, so it is only encapsulated on Windows platform.

2.1. Gene Sequences Conversion Tools Seqret. *Seqret* is mainly used to transfer the sequence files with different formats. For example, if the data provided by the user needs to be processed by *phylip*'s software encapsulation, but the *phylip*'s software encapsulation only supports the sequence file with *.phy* format, then the transfer of the file sequence's format is necessary. Here, it needs to call *seqret.exe*. Next, the parameters of *seqret* are introduced.

Seqret has two main parameters: one parameter is the sign of the input file format and the file name and the other is the sign of the output file format and the file name.

The described file manner is required to correspond with the description format specified in Uniform Sequence Address (USA). The USA formats are described as follows.

(i) "*file*": name of the input file. File is a sequence file with *.seq* as extension name.

(ii) "*file:entry*": combination form of file name and sequence ID.

(iii) "*format::file*": combination form of input file's organization format and name.

(iv) "*format::file:entry*": combination form of file's organization format, name, and index ID.

(v) "*database:entry*": combination form of database name and index ID.

(vi) "*database*": database name.

(vii) "*@file*": read each line in the file as an input sequence.

Seqret can recognize the form "*format::file*." For example, if we have a sequence file *file.seq* of the fasta type, we can express it as "*fasta::file.seq*."

In addition, *seqret* still contains two senior parameters: -feature shows the characteristics information of the sequence applied, and *-firstonly* indicates that the program terminates after reading a sequence from the sequence file.

2.2. ORF Search Tools Getorf. *Getorf* is used to find the *ORF* in the known RNA sequence and translate the obtained polypeptides.

The parameters of *getorf* are as follows.

(1) Input sequence file: nucleic acid sequence that corresponds with the USA formats.

(2) Output sequence file: gene sequence file that includes the *orf* search results.

(3) Senior options: *-circular* indicates whether the gene sequence is a ring, *-reverse* indicates whether to find ORF in the gene's completely reverse sequence, and *-flanking* indicates choosing a chain of branched gene sequence between the beginning and ending codons.

(4) Additional limited options: *-minsize*, *-maxsize*, *-find*, and *-table*.

  (a) *-minsize* indicates that the program needs to search a peptide sequence with the length not less than minisize.

  (b) *-maxsize* indicates that the program needs to search a peptide sequence with the length not more than maxisize.

  (c) *-find* is followed with digital options. The meaning of specific number is described in Table 1.

  (d) *-table* is followed with menu number from 0 to 23 to represent the organism's types. Here we do not describe the interpretation of specific number in detail.

2.3. Molecular Clock Based Maximum Likelihood Estimation Tools Proml. *Proml* is mainly used to construct amino acid sequence tree based on molecular clock maximum likelihood estimation. *Proml* has one input parameter which is a sequence file with *phy* format, and the sequence file includes numbers of amino acid sequences. Although the input parameter of *Proml* is very simple, but it has complex parameters settings, here we leave out the specific set options list.

Table 1: Interpretation of *-find's* digital options.

Digital option	Interpretation
0	Translate the *orf* between adjacent end codons
1	Output the *orf* between the beginning and ending codons
2	The nuclear sequences between end codons
3	The nuclear sequences between the beginning and end codons
4	The nucleosides side linking with the beginning codon
5	The nucleosides side linking with the start end codon
6	The nucleosides side linking with the terminal end codon

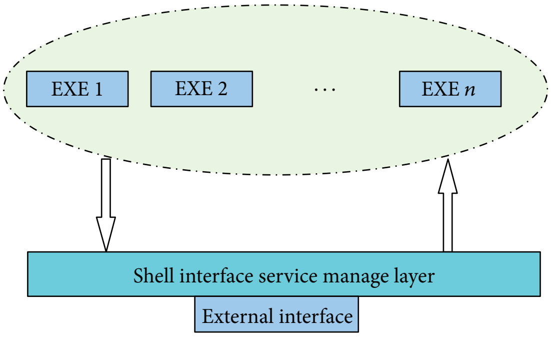


Figure 1: Encapsulation of the bioinformatics software.

3. Framework of Bioinformatics Software's Encapsulation and Publication

Firstly, according to the characteristics of bioinformatics software, we extract a unified interface to make it easy to integrate a lot of bioinformatics software. Thus, we encapsulate a layer of shell over the bioinformatics software, just as described in Figure 1. The shell program exposes a simple interface to the outside, thus making it convenient to publish the service.

Secondly, we embed the encapsulated bioinformatics software in the compiled GRAM service [9]; GRAM called this application and provided bioinformatics software's service to the outside [10]. The publishing part usually uses the web service. In the publishing interface, we premise that the users already know the existence of GRAM service, so that we can use the API that is provided by GRAM service to call the application sources that the GRAM contains; then bioinformatics software's publishing is realized.

Therefore, we form the framework of bioinformatics software's encapsulation and publishing, as Figure 2 shows.

As shown in Figure 2, in the upper level of our encapsulated software, GRAM encapsulates another layer, namely, the GRAM component internal service layer. The software is published with GRAM service's publication. During the publication, we use the API that is provided by GRAM to publish the bioinformatics software's function to the users and finally realize the bioinformatics software's publication.

3.1. Software Encapsulation on Windows Platform. As the software that will be encapsulated is all executable files, we create process to execute the exe files and deal with the interactive process by redirecting the standard input and output. During the encapsulation process, the main part is the application of redirection technology.

The specific redirecting process is as follows: we assume that there are two anonymous pipelines, two one-way pipelines: pipeline A and pipeline B; each pipeline has one input terminal and one output terminal.

First step: if we want to execute a command, we need to put this command to the execution file's process. We use *hStdInput* to stand for standard input; it is originally responsible for receiving the user's input from the keyboard; here we hang it up on pipeline A's output terminal and make it responsible for receiving pipeline A's output data.

Second step: now we have connected pipeline A's output terminal to the input terminal of execution file's process; namely, pipeline A's output terminal is execution file's input terminal, so that, if we write a command to pipeline A's input terminal, the execution file can get our command through pipeline A.

Third step: as known, it is impossible for pipeline A to receive the output data of execution file, so we need another pipeline, pipeline B, to receive it. We use *hStdOutput* to stand for standard output, it is originally sent to the screen, and here we hang it up on pipeline B's input terminal and make pipeline B responsible for receiving exe file's output data. What is more, *hStdError* is standard error output, it is also originally sent to the screen, and we hang it up on pipeline B's input terminal too.

Forth step: now pipeline B's input terminal is connected to the output terminal of execution file's process, so that pipeline B's output terminal is bioinformatics software's output terminal; software can receive data from this terminal and send it to the users or use it for further judgment.

3.2. Software Encapsulation on Linux Platform. The first step: create two pipelines in the parent process: pipeline 0 and pipeline 1; each pipeline has two terminals, respectively, for reading and writing. For each pipeline, two file descriptors will be generated: one is used to read data from specific file and the other is used to write data to the specific file.

The second step: call *fork*() to create a new child process. So there are two pipelines for both the parent and child processes, including four descriptors, respectively, for reading and writing two specific files. The two specific files are, respectively, indicated by the four descriptors of the parent and child processes. The general situation is shown in Figure 3.

The third step: for pipeline 0 and pipeline 1, respectively, turn off one pipeline's reading terminal and the other pipeline's writing terminal between the parent and child processes. For example, turn off pipeline 0's reading terminal and pipeline 1's writing terminal in the parent process; accordingly, turn off pipeline 0's writing terminal, and pipeline 1's reading terminal in the child process. Thus, the parent process can write data to pipeline 0's file through its writing terminal, and then the child process can read its parent process's data from pipeline 0's reading terminal. And similarly, the child process's feedback information can be written to pipeline 1,

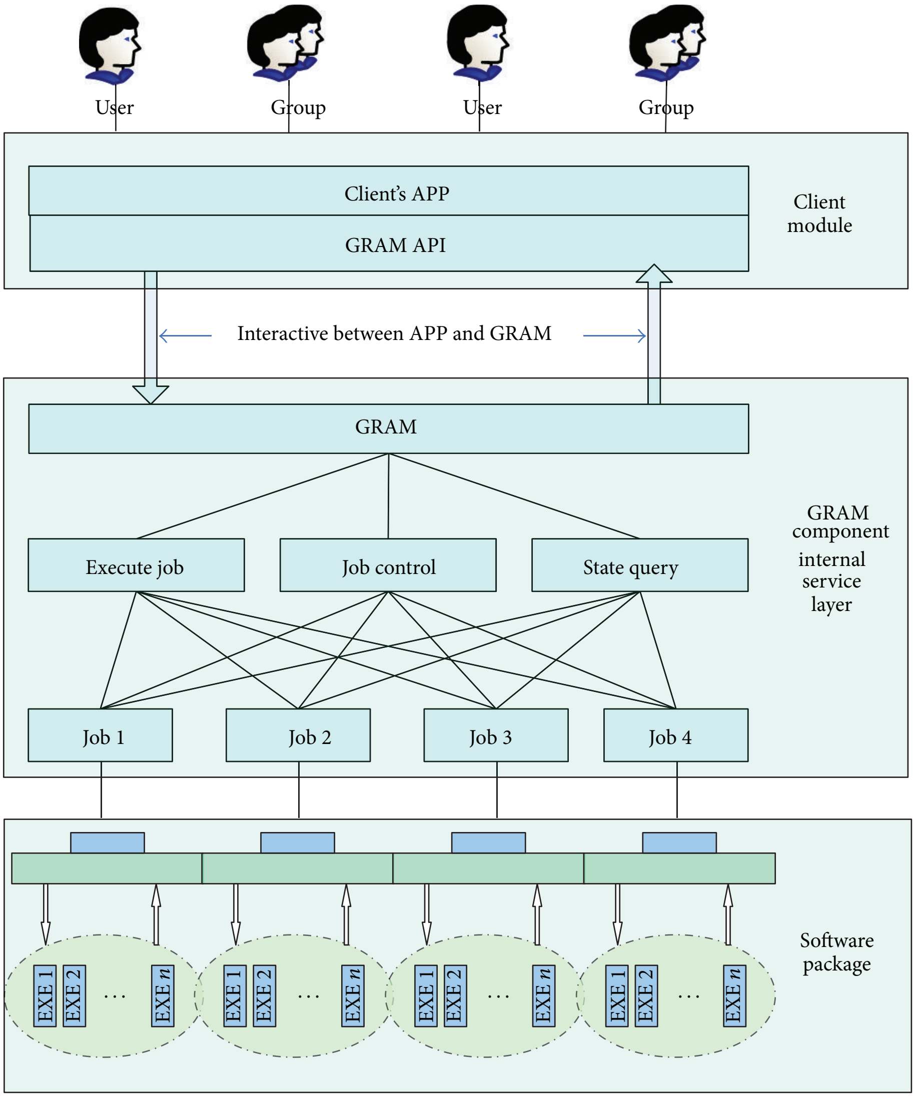


Figure 2: Framework of bioinformatics software's encapsulation and publication.

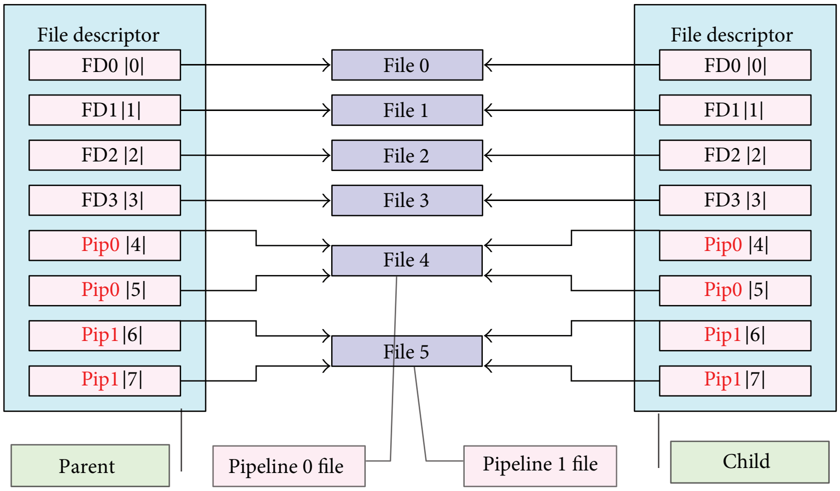


Figure 3: Mapping of parent and child process's file descriptor table.

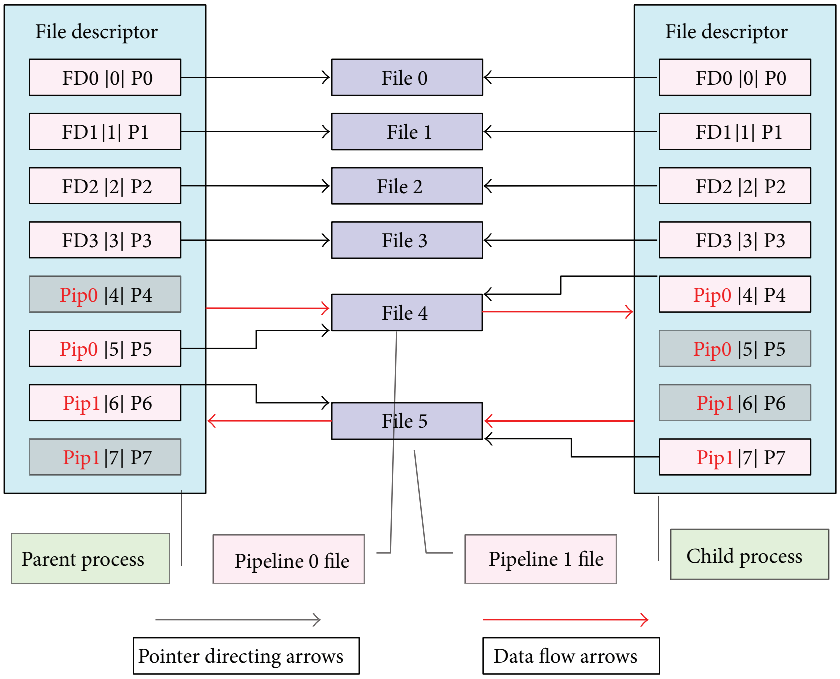


Figure 4: Piping communication schemes of the parent and child processes.

and then the parent process can receive the child process's information. So the communication between the child process and parent process is realized. The specific procedure is shown in Figure 4.

The forth step: we need to redirect the standard input/output of the child process. Here, we need to call function *dup2*() in the child process, change the file that the child process's standard input/output indicates by *dup2*(), and finally finish the redirecting of the standard input/output.

3.3. Bioinformatics Software's Publication. After encapsulating bioinformatics software, we should publish it next. Firstly, we further encapsulate the software with a GRAM service provided by Globus. Secondly, we publish the GRAM service. By these two steps the bioinformatics software is published.

We compile a client application by using GRAM component's API, and the users can call the bioinformatics software through this application. Next, we introduce the client's realization in detail.

Firstly, we take a look at the GRAM API that Globus project team [11] provides to us. Globus project team publishes an encapsulation named *org.globus.gram*, and this encapsulation realizes all the necessary API functions that are needed when calling the GRAM function. We mainly call the functions of class *Gram* and *GramJobListener* to complete the client program and realize the process that submitting remote services through the GRAM components.

Secondly, we consider the specific process to submit services. When the users call GRAM services' specific application, the main task is to finish the compiling of the resource description file, namely, forming an RSL file. RSL is a cloud resource description language based on XML language. RSL defines various kinds of labels to describe the resource, methods, and details of the calling process.

The users' tasks can be divided into two kinds, namely, single job task and multiple jobs task. For example, the RSL file is a description file of single job task. A single job task contains only one job, while a multiple jobs task contains numbers of jobs.

The submitting modes of user's tasks can be also divided into two kinds, namely, batch mode and no batch mode. In batch mode, the application program will be blocked after the user submits tasks and return after the tasks are completed and the results are returned, while, in no batch mode, the application program returns after the user submits tasks, so that the user can continue to deal with other tasks. If the user wants to observe the specific conditions of the submitted task, he can query the task's status by calling the task examination management interface provided by GRAM.

The XML documents form the standards and principles of the communication between applications. We united describe the communication content between the client and the server through XML documents, and the specific communication form is described in Figure 5.

As shown in Figure 5, line 1 shows that client program sends the information that the user requires to the stub module called by the client. Line 2 shows that the stub module encapsulates the information into standard format according to the provision way and measure and sends the encapsulated information to the server stub module. Line 3 shows that the server stub module analyzes the received information and gets the information that the user demands and then sends this information to the service realization program, so that the program can deal with the user's requirements. Line 4 shows that the service realization program sends the processed information back to the stub module. Line 5 shows that the server stub module encapsulates the processed

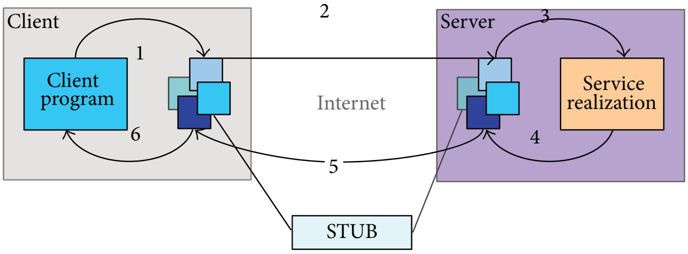


Figure 5: Description of the communication between client and server.

information and sends it to the client stub module. Line 6 shows that the client stub module analyzes the information and feeds it back to the client program.

4. Experiments

4.1. Encapsulation Interface Test on Windows Platform. First of all, we test the correctness of the bioinformatics software on Windows platform. We encapsulate three application programs, namely, *seqret.exe*, *getorf.exe*, and *proml.exe*. As described in the design, the encapsulation interface has three required parameters, including the file name, the input file, and the output file, and two optional domains, namely, the static input parameter and the dynamic input parameter.

Firstly, we test the correctness of *seqret's* encapsulation with the three required parameters; the result is shown in Figure 6. Obviously, the test is successful.

After calling *seqret's* encapsulation, the corresponding result file *seqret-out.phy* is generated in the directory from which the program runs, and the corresponding log file is also generated in the LogFile folder. The result is shown in Figure 7.

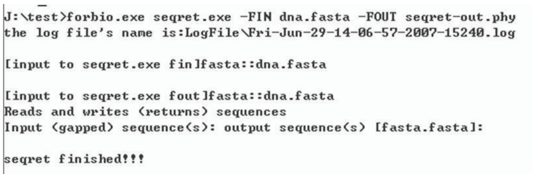
Next, we test *proml's* encapsulation. As *proml* is an application program on Windows platform, so we only test its encapsulation on Windows platform. *Proml* chooses the interactive parameters to communicate with the users. In our test, we choose the parameter –I and import three interactive parameters: u, 5, and y.

The test results are shown in Figure 8.

The log files record the implementation details of this software's calling process successfully, and its format is the same to the log file of *seqret*; here we leave out the details of the log file.

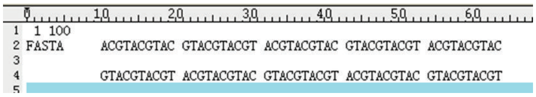
4.2. GT's Local Task Submitting Test on Linux Platform. The operating system we use is Red Hat Enterprise Linux Advanced Server 4 [12], and the GT's version is Globus Toolkit 4.0.2 [13]. In the submitting test, the description file is shown in Figure 9.

Firstly, we submit the task by using the command globus-run-ws of GT's command line tool. Before the submit, we need to generate an agent with the command grid-proxy-init firstly; this is because the agent can help do some necessary operation when the GRAM calls other remote file transfer tasks, and the user's certification is needed to be identified. Therefore, the user's certification is the precondition of GRAM components' application.



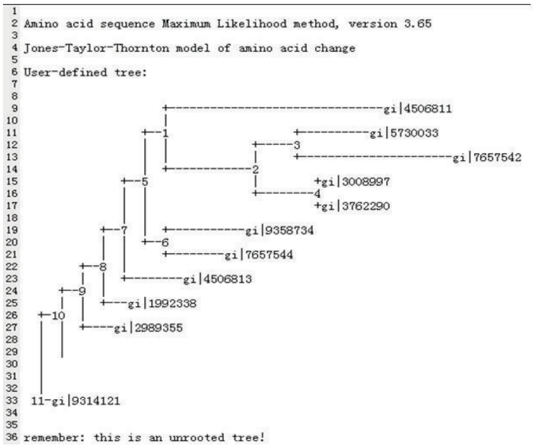
```
J:\test>forbio.exe seqret.exe -FIN dna.fasta -FOUT seqret-out.phy
the log file's name is:LogFile\Fri-Jun-29-14-06-57-2007-15240.log

[input to seqret.exe fin]fasta::dna.fasta

[input to seqret.exe fout]fasta::dna.fasta
Reads and writes (returns) sequences
Input (gapped) sequence(s): output sequence(s) [fasta.fasta]:

seqret finished!!!
```

Figure 6: Schematic diagram of calling *seqret's* encapsulation Windows platform.



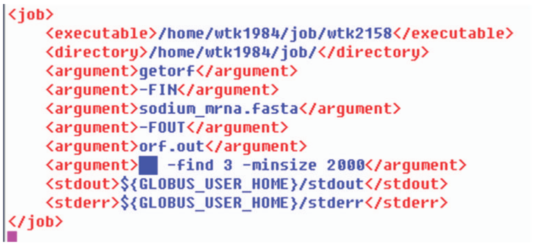
```
 1 100
FASTA      ACGTACGTAC GTACGTACGT ACGTACGTAC GTACGTACGT ACGTACGTAC

           GTACGTACGT ACGTACGTAC GTACGTACGT ACGTACGTAC GTACGTACGT

```

Figure 7: Result of the sequence's transformation.



```

Amino acid sequence Maximum Likelihood method, version 3.65

Jones-Taylor-Thornton model of amino acid change

User-defined tree:


                    +-------------------------------gi|4506811
                    |
                 +--1                  +----------gi|5730033
                 |  |             +----3
                 |  |             |    +----------------------gi|7657542
                 |  +-------------2
              +--5                |       +gi|3008997
              |  |                +-------4
              |  |                        +gi|3762290
              |  |
           +--7  |  +-----------gi|9358734
           |  |  +--6
           |  |     +--------gi|7657544
        +--8  |
        |  |  +--------gi|4506813
     +--9  |
     |  |  +--gi|1992338
  +-10  |
  |  |  +---gi|2989355
  |  |
  |  |
  |
  |
  |
 11-gi|9314121


remember: this is an unrooted tree!
```

Figure 8: Result of file of *proml.*



```
<job>
    <executable>/home/wtk1984/job/wtk2158</executable>
    <directory>/home/wtk1984/job/</directory>
    <argument>getorf</argument>
    <argument>-FIN</argument>
    <argument>sodium_mrna.fasta</argument>
    <argument>-FOUT</argument>
    <argument>orf.out</argument>
    <argument>■ -find 3 -minsize 2000</argument>
    <stdout>${GLOBUS_USER_HOME}/stdout</stdout>
    <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
</job>
```

Figure 9: Task description file.

```
[wtk1984@freedom  ~]$  ll  job/*orf.out  job/LogFile/*.log
-rw-r--r--  1  wtk1984 wtk1984      722 Jun 29 22:44  job/LogFile/Fri-Jun-29-14:44:15-
2007-93250.log
-rw-r--r--  1  wtk1984  wtk1984  1044826  Jun 29 22:44  job/orf.out
```

Box 1

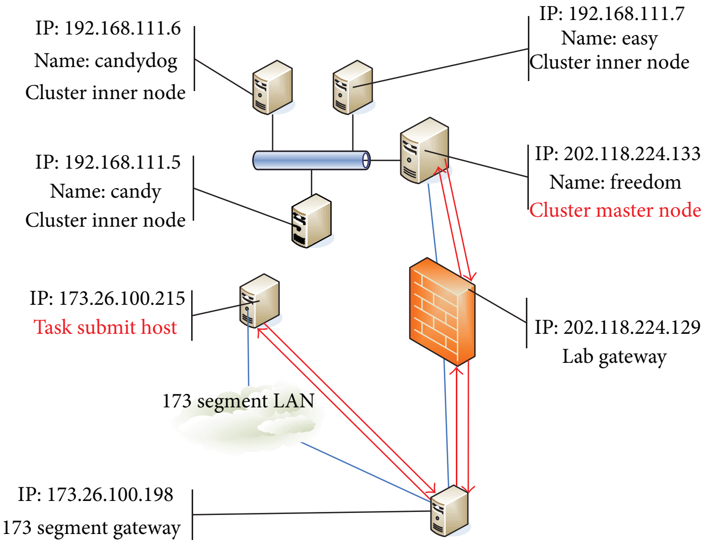


Figure 10: Frame of the network on which the test is conducted.

Next, we take the task description file shown in Figure 9 as one of the parameters we input in the command line to submit our task. In the task description file, we call *getorf* and input parameters *sodium_mrna.fasta, orf.out*, and -find 3-minsize 2000 as static parameters.

The result file and log file are as shown in Box 1: as the red front shows, the result file and log file are generated successfully. We check the records in the two files, and they are both correct.

4.3. Client Remote Calling Test. We divide the test types of this part into five kinds: 0 cloud node and its service query, 1 remote submitting of nonbatch mode and single job task, 2 remote submitting of batch mode and single job task, 3 remote submitting of batch mode and multiple jobs task, and 4 query of the status of the task that is submitted with batch mode. Then we show the task submitting test of types 0, 1, 3, and 4 in detail.

Before the test, we describe the logic of the network on which the test is conducted. Its frame is shown in Figure 10.

The red font shows the task submitting node and the cluster master node, the place where our task is processed. Our encapsulated software is stored in the master node. The red arrows indicate the flow of the information.

We test the task's submitting with test type 1, call *seqret* job, and process the gene sequence transfer job. Figure 11 shows the process of this test.

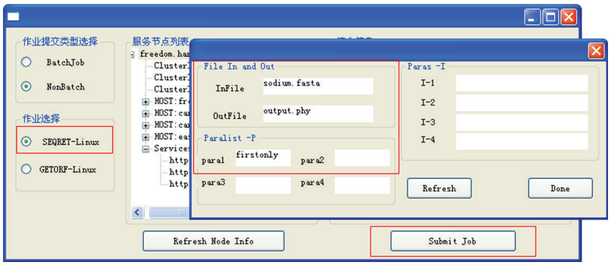

作业提交类型选择
○ BatchJob
● NonBatch
作业选择
● SEQRET-Linux
○ GETORF-Linux
File In and Out
InFile: sodium.fasta
OutFile: output.phy
Paralist -P
para1: firstonly
para2:
para3:
para4:
Paras -I
I-1:
I-2:
I-3:
I-4:

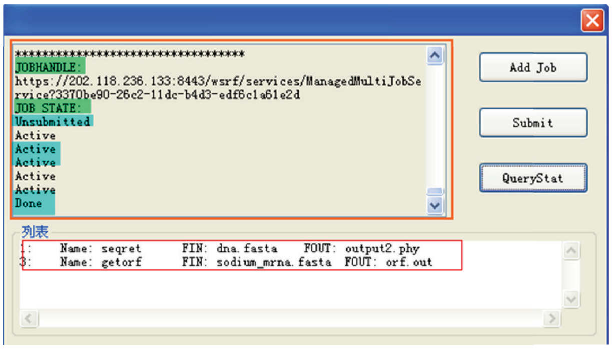
Figure 11: Submitting of no batch mode and single job task.

As Figure 11 shows, we can choose the task and their modes in the most left red box, set the task's parameters through the popup dialog box, and then submit the task by Submit Job button. As the submitted task is nonbatch, the application program blocks after the submitting until the remote execution is finished, and the completion signal is returned.

Next, we login the server and check the result file and the log file; they are both correct. Here, we leave out the details of the inspection.

Finally, we test the task's submitting with test type 3. We check the task's status by calling the query function.

As shown in Figure 12, the list in the bottom shows the tasks' name and parameters in detail, and the upper windows shows the results of the query; from it we can see two items:



```
**********************************
JOBHANDLE:
https://202.118.236.133:8443/wsrf/services/ManagedMultiJobSe
rvice?3370be90-26c2-11dc-b4d3-edf6c1a61e2d
JOB STATE:
Unsubmitted
Active
Active
Active
Active
Active
Done
```

```
1:    Name: seqret      FIN: dna.fasta        FOUT: output2.phy
3:    Name: getorf      FIN: sodium_mrna.fasta  FOUT: orf.out
```

Figure 12: Query of the task's status.

the job handle and the job state. We can see that our submitted task experiences the process of *Unsubmitted->Active->Done*.

5. Conclusion

According to the secure encapsulation and publishing of bioinformatics software, this paper introduces the methods of encapsulating and publishing the existing services in the cloud environment and realizes a prototype system to publish bioinformatics software in the grid and cloud computing environment.

In the publishing part, the main process is the analysis of the application software's business and data flow. During the analysis, according to the interaction between processes, we use the communication mechanism of the processes to simulate the man-machine interaction by the application of the pipeline's redirection technology. Finally, according to the results of the analysis, we summarize the characteristics of bioinformatics software's external interface and make the interface simple and universal.

We use the services provided by cloud development tools in the publishing process, write interface processing program for specific application programs, and publish it with corresponding publishing mechanism. Finally, we combine the remote calling of bioinformatics software with cloud environment and form the prototype system of the biological cloud.

Acknowledgments

This work was supported by the National Basic Research Program of China under Grant no. G2011CB302605, the National Natural Science Foundation of China (NSFC) under Grant no. 61173145, and the National High Technology Research and Development Program of China under Grant no. 2011AA010705.

References

[1] D. Gilbert, "Bioinformatics software resources," *Briefings in Bioinformatics*, vol. 5, no. 3, pp. 300–304, 2004.

[2] C. Oehmen and J. Nieplocha, "ScalaBLAST: a scalable implementation of BLAST for high-performance data-intensive bioinformatics analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 740–749, 2006.

[3] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.

[4] F. Joseph, *Inferring Phylogenies*, Sinauer Associates, 2003.

[5] P. Rice, L. Longden, and A. Bleasby, "EMBOSS: the european molecular biology open software suite," *Trends in Genetics*, vol. 16, no. 6, pp. 276–277, 2000.

[6] Y.-L. Chen, B.-C. Cheng, H.-L. Chen et al., "A privacy-preserved analytical method for ehealth database with minimized information loss," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 521267, 9 pages, 2012.

[7] J. Chen, F. Qian, W. Yan, and B. Shen, "Translational biomedical informatics in the cloud: present and future," *BioMed Research International*, vol. 2013, Article ID 658925, 8 pages, 2013.

[8] R. de Paris, F. A. Frantz, O. N. de Souza, and D. D. Ruiz, "wFReDoW: a cloud-based web environment to handle molecular docking simulations of a fully flexible receptor model," *BioMed Research International*, vol. 2013, Article ID 469363, 12 pages, 2013.

[9] B. Sotomayor, *GT4 Programmer's Tutorial*, Globus Toolkit Develope Team, 2004.

[10] L. Ferreira and V. Berstis, *Introduction to Grid Computing with Globus*, IBM Redbooks, 2003.

[11] J. Pedraza, M. A. Patricio, A. de Asís, and J. M. Molina, "Privacy and legal requirements for developing biometric identification software in context-based applications," *International Journal of Bio-Science and Bio-Technology*, vol. 2, no. 1, pp. 13–24, 2010.

[12] D.-Y. Lee, S. Bae, J. H. Song et al., "Self-Organized Software Platform (SOSp)-based mobile chronic disease management with agent-based HL7 interface," *International Journal of Bio-Science and Bio-Technology*, vol. 5, no. 1, pp. 59–72, 2013.

[13] H. Kim, Y. Kim, and P. Lee, "Reconfiguration mechanisms for virtual organization using remote deployment of grid service," *International Journal of Grid and Distributed Computing*, vol. 2, no. 1, pp. 27–38, 2009.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 329046, 6 pages
http://dx.doi.org/10.1155/2013/329046

Research Article

Image Analysis of Endosocopic Ultrasonography in Submucosal Tumor Using Fuzzy Inference

Kwang Baek Kim[1] **and Gwang Ha Kim**[2]

[1] *Department of Computer Engineering, Silla University, Busan 617-736, Republic of Korea*
[2] *Department of Internal Medicine, Pusan National University School of Medicine and Biomedical Research Institute, Pusan National University Hospital, Busan 602-739, Republic of Korea*

Correspondence should be addressed to Gwang Ha Kim; doc0224@pusan.ac.kr

Received 2 June 2013; Accepted 8 July 2013

Academic Editor: Sabah Mohammed

Endoscopists usually make a diagnosis in the submucosal tumor depending on the subjective evaluation about general images obtained by endoscopic ultrasonography. In this paper, we propose a method to extract areas of gastrointestinal stromal tumor (GIST) and lipoma automatically from the ultrasonic image to assist those specialists. We also propose an algorithm to differentiate GIST from non-GIST by fuzzy inference from such images after applying ROC curve with mean and standard deviation of brightness information. In experiments using real images that medical specialists use, we verify that our method is sufficiently helpful for such specialists for efficient classification of submucosal tumors.

1. Introduction

Recently, many digestive diseases are found in the early stage due to increasing usage of upper gastrointestinal endoscopy. One of those disease groups is a submucosal tumor (SMT). An SMT is a spherical or hemispheric lesion projected toward the lumen of which main lesion exists below the mucosa and its surface is covered with normal gastrointestinal mucosa. Among those SMTs, leiomyoma, cyst, fibroma, lipoma, and hemangioma are benign tumors, but gastrointestinal stromal tumor (GIST), leiomyosarcoma, and lymphoma have malignant potential.

Because native-eye findings of the endoscopic image are very similar and histological confirmation is mainly not possible by endoscopic biopsy only, medical specialists have great difficulty in classifying them correctly. Endoscopic ultrasonography (EUS) overcomes such difficulty in diagnosing SMTs, and it is also used for staging malignant tumors in the digestive tract [1, 2].

Most SMTs are benign. However, benign SMTs are not easily distinguished from malignant SMTs, and even if they are truly benign, there is no agreement among specialists in how frequently a followup is needed or when operative treatment should be given to the patient.

GISTs have a risk of metastatic relapse, especially in the peritoneum and liver, after surgery for localized diseases [3, 4]. Therefore, every GIST is now considered as potentially malignant, and so all GISTs may need to be resected, even small intramural lesions of the stomach [5].

In practice, the differentiation of GISTs from benign SMTs is essential to clinical management. However, the studies for distinguishing between GISTs and other benign mesenchymal tumors by EUS are still only a few [6, 7].

In addition, there are limitations in the analysis of the characteristic EUS features because of poor interobserver agreement by subjective interpretation of EUS images [8, 9]. Therefore, if an objective analysis for EUS images would be possible especially by means of computer-assisted image analysis, the previous limitation might be overcome.

Thus, in this paper, we propose a method to extract areas of GIST and lipoma automatically from the standardized ultrasonic image to assist those endoscopists. We also propose an algorithm to differentiate GIST from non-GIST by fuzzy inference [3] from such images after applying an ROC

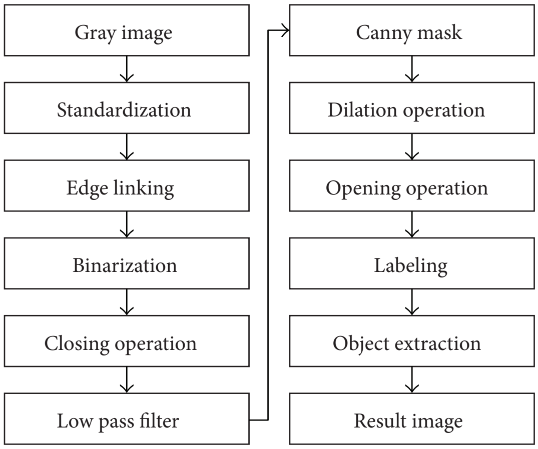
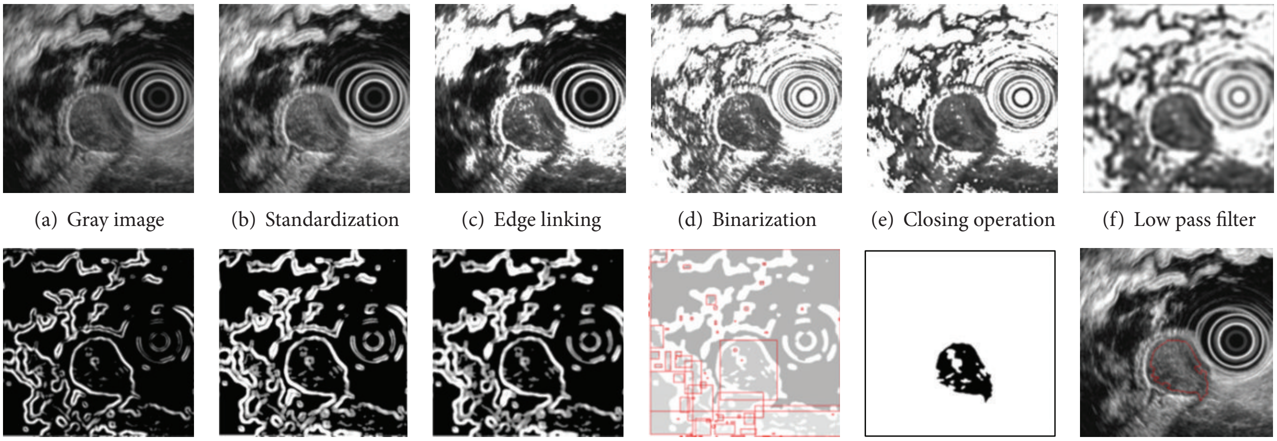


Figure 1: Process for extracting GIST.



(a) Gray image (b) Standardization (c) Edge linking (d) Binarization (e) Closing operation (f) Low pass filter

(g) Canny mask (h) Dilation operation (i) Opening operation (j) Labeling (k) Object extract (l) Result image

Figure 2: GIST extraction.

curve with mean and standard deviation of the brightness information.

2. Materials and Methods

2.1. Extracting Gastrointestinal Stromal Tumor (GIST). EUS was performed using a radial scanning ultrasound endoscope (GF-UM2000; Olympus, Tokyo, Japan) at 7.5 MHz. All the examinations were performed under intravenous conscious sedation (midazolam with or without meperidine). Scanning of the tumor was performed after filling the stomach with 400–600 mL of deaerated water. At least 10 endosonograms were recorded for each lesion, and these images were digitally saved in the Windows bitmap format.

Reviewing the EUS images was performed by a single experienced endosonographer (Kim et al. [6]) who was kept "blinded" to the final diagnosis, and only one highest quality EUS image for each lesion was selected for further analysis, which was performed on a standard desktop computer.

GIST is a mesenchymal tumor with malignant potential found in the stomach, and small and large intestine. The majority (60~70%) is found in the stomach. Therefore, we included gastric GISTs located in this study.

Figure 1 shows the overall process for extracting GIST by the proposed method.

There are too many edges in the GIST area from the standardized EUS image [10–12], but the boundary lines could be removed according to the characteristic that boundary lines have too high or too low brightness.

For pixels that have sufficient brightness (experimental threshold above 30), we apply an edge linking method that connects the current pixel to adjacent pixels if formula (1) is satisfied. The experimental threshold Th in our study was 130:

$$\left|\Delta G\left(x,y\right) - \Delta\left(G'\left(x',y'\right)\right)\right| \le \text{Th}. \tag{1}$$

Then we remove low brightness pixels in the GIST area by setting those pixels' brightness as 255 if the brightness is no higher than 40. Figure 2(d) shows the result of the proposed procedures.

Also, noise removal is followed by applying a morphological closure operation in order to fill the gap or little holes while maintaining the size and the shape of the object

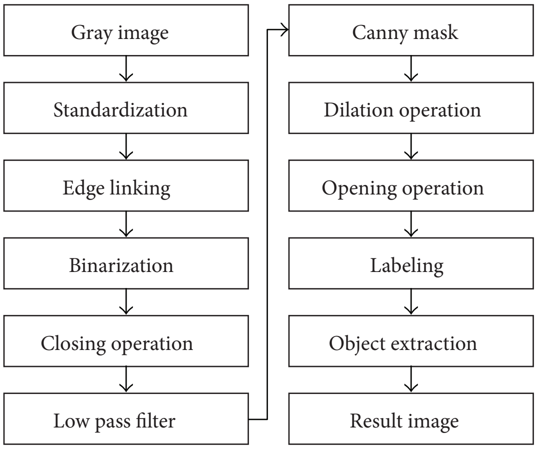
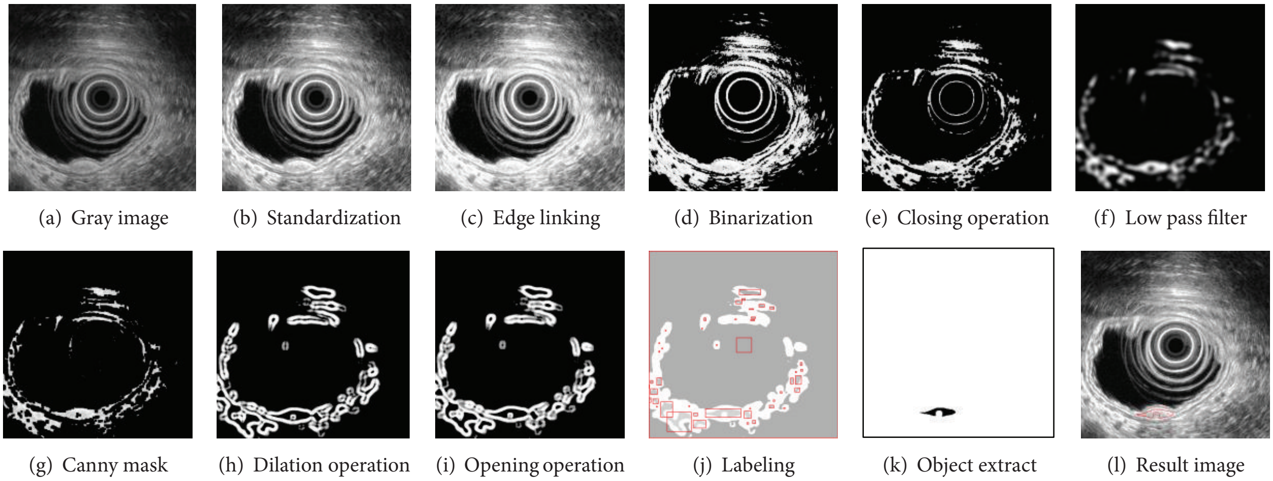


Figure 3: Process for extracting lipoma.



(a) Gray image (b) Standardization (c) Edge linking (d) Binarization (e) Closing operation (f) Low pass filter

(g) Canny mask (h) Dilation operation (i) Opening operation (j) Labeling (k) Object extract (l) Result image

Figure 4: Lipoma extraction.

as shown in Figure 2(e). The resulting image is smoothed by a Butterworth low-frequency filter for irregular edges in the tumor area as shown in Figure 2(f). Boundary lines are extracted by using a noise-insensitive Canny mask to remove minute noise. From that noise-free Figure 2(g), we apply a dilation operation to reconnect unexpectedly disconnected boundary lines during the preceding process and also an opening operation to maintain the original size of objects after such noise removal as shown in Figure 2(i).

Then we apply a GrassFire algorithm to label them as shown in Figure 2(j) as all pixels in the same object have the same identification number and remove objects including lens and other subtle noise. Finally, the GIST area is extracted by taking objects that have high density of pixels as demonstrated in Figure 2(l) to finish the process.

2.2. Extracting Lipoma. Lipoma is a well-capsulated benign tumor consisting of matured adipocyte. It can be found everywhere but it is usually seen in the subcutis of normal adipose tissue such as thigh, arm, and torso. Lipoma is one of the most frequent benign tumors found in the soft tissue among age 40~60, and it is often found in the stomach during endoscopy.

Lipoma area usually has high brightness, and its boundaries are clear. We apply histogram smoothing to regulate brightness distribution of lipoma area. We control lower brightness of pixels (<75) as giving brightness zero in order to remove dark noise. 75 is an experimental threshold that is the lowest brightness of lipoma area.

The process of noise removal and object extraction is similar to that of the GIST case explained in Section 2.1. Figures 3 and 4 demonstrate the diagram and corresponding treated images.

2.3. Classifying Tumor by Fuzzy Inference. In our method, an endoscopist chooses the tumor area on the standardized EUS image. Then we apply the average brightness (MEAN) and standard deviation (SD) information to the ROC curve [13, 14] which visualizes the prediction rate of true positivity and false positivity. The results are used as the membership function intervals of our fuzzy theory [15, 16]. By applying the ROC curve to the MEAN and SD of the chosen tumor area,

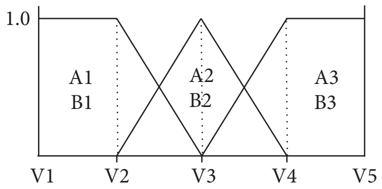


Figure 5: First half membership functions.

Table 1: Results for the ROC curve.

	Sensitivity	1-specificity	AUC
Mean	65.12	0.896	0.091
SD	74.97	0.917	0.273

we obtain Table 1, and those results are used to establish fuzzy membership function intervals.

Figure 5 denotes the membership functions where A denotes MEAN and B denotes SD.

In Figure 5, intervals of V1, V2, . . ., V5 are categorized as reported in Table 2.

After computing the degree of membership using Figure 5, we use fuzzy inference rules as follows.

IF A1 and B1 then G1
IF A1 and B2 then G1
IF A1 and B3 then G1
IF A2 and B1 then G1
IF A2 and B2 then G2
IF A2 and B3 then G2
IF A3 and B1 then G2
IF A3 and B2 then G2
IF A3 and B3 then G3.

We use popular Min_Max reasoning [17] and apply it to the membership degree of GIST as shown in Figure 6 and use centroid method as a defuzzifier as shown in formula (2). Finally, the class of tumor is defined by criteria as shown in Table 3:

$$W_z = \frac{\sum \mu(X_i) X_i}{\sum \mu(X_i)}. \qquad (2)$$

3. Results

In experiments, we used real EUS images from endoscopists for three different types of tumor, ten cases per type. The software is written in VC++ 2005 on notebook with Intel Pentium dual-core 2 GHz CPU and 3 GB RAM.

The result of GIST and lipoma extraction is shown in Table 4. And Figure 7 demonstrates successful extraction cases of GIST and Lipoma.

Two failed cases for GIST are due to unexpectedly high density of pixels of noises, and one failed case in extracting lipoma is due to including unnecessary objects when we extract edges with the Canny mask. Such cases are shown in Figure 8.

Table 5 shows the classification results of three different tumors by fuzzy inference. We take this as two-class problem in that we are only interested in GIST and non-GIST. There exist one or two failed cases for each class but overall, it is sufficiently accurate for endoscopists as an auxiliary tool.

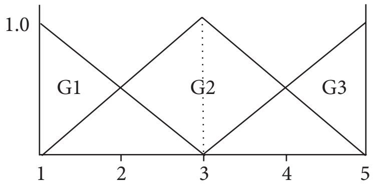


Figure 6: Membership functions for tumor classification.

Table 2: Membership function intervals.

	V1	V2	V3	V4	V5
			Mean		
A1	0	55	65		
A2		55	65	75	
A3			65	75	255
			SD		
B1	0	65	75		
B2		65	75	85	
B3			75	85	255

Table 3: Criteria for tumor classification.

$1 \leq W_z \leq 2$	Non-GIST (cyst)
$2 \leq W_z \leq 4$	GIST
$4 \leq W_z \leq 5$	Non-GIST (lipoma)

Table 4: Tumor extraction results.

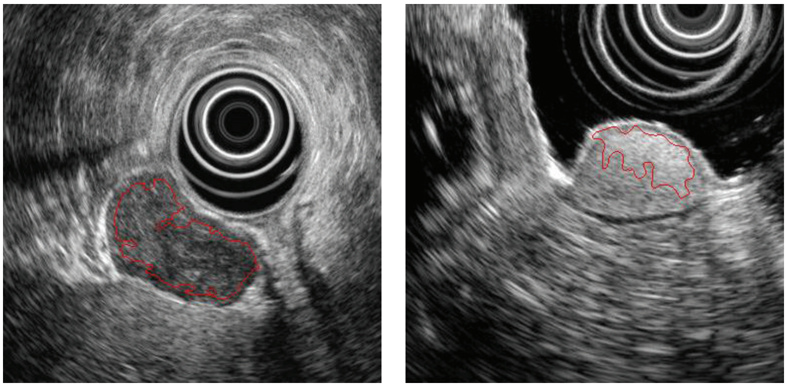
	Successful/total
GIST	8/10
Lipoma	9/10

Table 5: Results of fuzzy analysis.

	Successful/total
GIST	8/10
Non-GIST	19/20

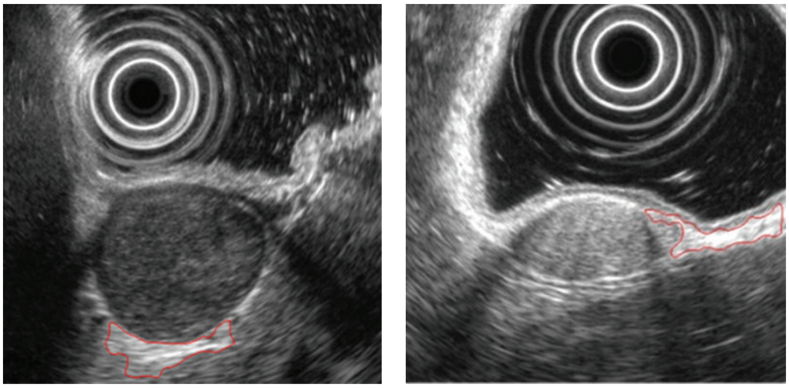
4. Conclusions

In this paper, we propose a method to extract GIST and lipoma from EUS images and a classification scheme with fuzzy inference whether it is a GIST or not. From the standardized EUS images, we apply various image processing algorithms such as binarization, morphological operations, GrassFire algorithm, Canny mask, smoothing, and so forth. in order to remove noise. Then a target tumor is extracted by the characteristic of high density of pixels. We also propose



(a) Correct GIST image (b) Correct lipoma image

Figure 7: Correct tumor extraction.



(a) Incorrect GIST image (b) Incorrect lipoma image

Figure 8: Incorrect tumor extraction.

a method to discriminate GIST from non-GIST tumor with fuzzy inference rules.

In experiments which used real clinical data, the extraction of GIST and lipoma is not yet fully successful; the accuracy is about 85%. However, the classification of tumors is almost correct overall, where 27 of 30 cases have been correctly classified. This experience stimulates us to develop more accurate extraction algorithm in the future.

Acknowledgments

This study was supported by a Grant from the National R&D Program for Cancer Control, Ministry for Health, Welfare and Family affairs, Republic of Korea (0920050).

References

[1] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[2] P. May, H.-C. Ehrlich, and T. Steinke, "ZIB structure prediction pipeline: composing a complex biological workflow through web services," in *Euro-Par: Parallel Processing*, W. E. Nagel, W. V. Walter, and W. Lehner, Eds., vol. 4128 of *Lecture Notes in Computer Science*, pp. 1148–1158, 2006.

[3] J.-Y. Blay, S. Bonvalot, P. Casali et al., "Consensus meeting for the management of gastrointestinal stromal tumors. Report of the GIST Consensus Conference of 20-21 March 2004, under the auspices of ESMO," *Annals of Oncology*, vol. 16, no. 4, pp. 566–578, 2005.

[4] M. Miettinen, L. H. Sobin, and J. Lasota, "Gastrointestinal stromal tumors of the stomach: a clinicopathologic, immunohistochemical, and molecular genetic study of 1765 cases with long-term follow-up," *American Journal of Surgical Pathology*, vol. 29, no. 1, pp. 52–68, 2005.

[5] C. D. M. Fletcher, J. J. Berman, C. Corless et al., "Diagnosis of gastrointestinal stromal tumors: a consensus approach," *Human Pathology*, vol. 33, no. 5, pp. 459–465, 2002.

[6] G. H. Kim, D. Y. Park, S. Kim et al., "Is it possible to differentiate gastric GISTs from gastric leiomyomas by EUS?" *World Journal of Gastroenterology*, vol. 15, no. 27, pp. 3376–3381, 2009.

[7] T. Okai, T. Minamoto, K. Ohtsubo et al., "Endosonographic evaluation of c-kit-positive gastrointestinal stromal tumor," *Abdominal Imaging*, vol. 28, no. 3, pp. 301–307, 2003.

[8] M. F. Catalano, M. V. Sivak Jr., R. A. Bedford et al., "Observer variation and reproducibility of endoscopic ultrasonography," *Gastrointestinal Endoscopy*, vol. 41, no. 2, pp. 115–120, 1995.

[9] F. Gress, C. Schmitt, T. Savides et al., "Interobserver agreement for EUS in the evaluation and diagnosis of submucosal masses," *Gastrointestinal Endoscopy*, vol. 53, no. 1, pp. 71–76, 2001.

[10] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 1999.

[11] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *Proceedings of the 10th IEEE Interantionsl Symposium on High Performance Distributed Computing (HPDC '01)*, pp. 181–194, San Francisco, Calif, USA, August 2001.

[12] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the grid: an open grid services architecture for distributed systems integration," Tech. Rep., Global Grid Forum, 2002.

[13] K.-B. Kim, S. Kim, and G.-H. Kim, "Analysis system of endoscopic image of early gastric cancer," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 89, no. 10, pp. 2662–2669, 2006.

[14] K. B. Kim, "Submucosal tumor analysis of endoscopic ultrasonography images," *Journal of Korea Multimedia Society*, vol. 13, no. 7, pp. 1044–1050, 2010.

[15] P. Alli, P. Ramasubramanian, and V. Sureshkumar, "Oil spills detection In SAR images using nonlinear fuzzy filter," *International Journal of Advanced Science and Technology*, vol. 25, pp. 7–16, 2010.

[16] D. O. Aborisade, "Novel fuzzy logic based edge detection technique," *International Journal of Advanced Science and Technology*, vol. 29, pp. 75–82, 2011.

[17] A. Kandel and G. Langholz, *Fuzzy Control Systems*, CRC Press, 1994.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 372646, 7 pages
http://dx.doi.org/10.1155/2013/372646

Research Article

Designing a Bioengine for Detection and Analysis of Base String on an Affected Sequence in High-Concentration Regions

Debnath Bhattacharyya,[1] Bijoy Kumar Mandal,[1] and Tai-hoon Kim[2]

[1] *Department of Computer Science & Engineering, Faculty of Engineering and Technology, NSHM Knowledge Campus, Durgapur 713212, India*
[2] *Department of Convergence Security, Sungshin Women's University, 249-1, Dongseon-dong 3-ga, Seoul 136-742, Republic of Korea*

Correspondence should be addressed to Tai-hoon Kim; taihoonn@daum.net

Received 4 July 2013; Accepted 13 July 2013

Academic Editor: Sabah Mohammed

We design an Algorithm for bioengine. As a program are enable optimal alignments searching between two sequences, the host sequence (normal plant) as well as query sequence (virus). Searching for homologues has become a routine operation of biological sequences in 4 × 4 combination with different subsequence (word size). This program takes the advantage of the high degree of homology between such sequences to construct an alignment of the matching regions. There is a main aim which is to detect the overlapping reading frames. This program also enables to find out the highly infected colones selection highest matching region with minimum gap or mismatch zones and unique virus colones matches. This is a small, portable, interactive, front-end program intended to be used to find out the regions of matching between host sequence and query subsequences. All the operations are carried out in fraction of seconds, depending on the required task and on the sequence length.

1. Introduction

It is known that viroids are the smallest replicating pathogenic agents (see [1] for relevant references), which is entirely composed of RNA with genome sizes in the range of 330–380 nucleotides [2], that is 10 times smaller than the smallest bacteriophage of *Escherichia coli* [3]. It is also known that they infect a wide variety of plants and produce severe disease symptoms in many plants [4–12], but here is no evidence for the existence of a protective protein coat for viroids. The molecular mechanisms by which viroids replicate and interact with their hosts are not yet understood. In its most severe form, the disease [5, 6] caused by potato spindle tuber viroid (PSTV) causes general stunting of potato plant growth, deformity of the upper foliage, and production of disfigured potatoes [5]. Mild strains of PSTV which produce barely detectable symptoms have also been isolated [7]. Furthermore, plants infected with mild strains are somehow protected from developing symptoms following subsequent inoculation with severe strains [8, 9]. The sequence of the 247 nucleotide residues of the single strand circular RNA of avocado sunblotch viroid (ASBV) was determined using partial enzymes cleavage methods on overlapping viroid fragments obtained by partial ribonucleic digestion followed by ^{32}p-labelling *in vitro* at their $5'$-ends. ASBV is much smaller than potato spindle tuber viroid (PSTV; 359 residues) and chrysanthemum stunt viroid (CSV; 356 residues). The sequences of the viroid progeny and the cloned DNA were identical. *In vitro* mutagenesis of infectious PSTV cDNAs will allow systematic investigation of the role of specific sequences in viroid replication and pathogenesis [10]. A complex of considerable stability is possible between the $5'$-end of U1 RNA and a specific nucleotide sequence of the potato spindle tuber viroid complement. Small nuclear RNAs (snRNAs) that are associated with ribonucleoprotein particles are believed by some to be involved in the processing of the primary transcription products of split genes. The $5'$-end of one such RNA, U1, has been shown to exhibit complementarity with the ends of introns, and it is believed that this affords a mechanism ensuring correct excision of the intron sequences and accurate joining of the coding sequences [11]. The invention provides a novel retroviral packaging system, in which retroviral packaging constructs and packageable vector transcripts are produced from high-expression plasmids by replicating in a human's cell via the enzyme reverse transcriptase to produce DNA from its RNA genome. Retroviruses are enveloped viruses that belong to the viral family retroviridae. High titers of recombinant retrovirus are produced in infected cells. The methods of the invention include the use of the novel retroviral constructs to transduce primary human cells, including T cells and human hematopoietic stem cells, with foreign genes by cocultivation at high efficiencies. The invention is useful for the rapid production of high viral supernatants, and to transduce with high-efficiency cells that are refractory to transduction by conventional means [12].

2. Basis of the Algorithm

There are four issues which are focused mainly to provide for detection of a fixed base string on an affected sequence.

2.1. Similarity. To define similarity, perhaps it is useful to first introduce the notion of "distance" between two strings. The distance between two strings is zero if they are exactly the same. The distance between two strings increases if they get more dissimilar. One way of defining distance between two strings is to look at the amount of change they needed to do to one to obtain the other. They could go on to introduce other changes, insert, and delete. Insert "happens" when they inserted some letter into the sequence (at some position), and delete happens when they deleted some letter at some position.

2.2. Edit Distance. This is defined as the minimum number of changes to be performed on one sequence to make it exactly the same as another.

2.3. Alignment of Sequence. For every two sequences, there are huge permutations of possible alignments (cubic in the length of sequences). Alignment procedure itself can be visualized as a series of insert, delete operations.

2.4. Scoring Function. A scoring function determines this notion of goodness of alignment. They could compute the distance between alignments in such a way that the cost of a match is 0 (when the sequence on top and below has the same ith character). Cost of a mismatch is that they could choose different scoring schemes. Another sample scoring scheme could give lesser weights for replacement of A by T, and G by C (and vice versa) as against replacement of A by G or the others. Domain knowledge is used while determining scoring schemes.

3. Designing of the Algorithm

There are basic steps that constitute the whole process of analysis for high-concentration regions (HCR) detection of a fixed base string on an affected sequence and those steps are as follows.

3.1. Match Occurs in the following Way

$Q[i] = H[j]$ to $H[m - L + 1]$.

As for example, $Q[1] = H[1]$ first match found.

Next $Q[2]$ match with $H[1]$ to $H[m - L + 1]$.

This process will continue at the end of query sequence. This process is repeated at the end of query sequence, until all possible matches are found.

Match found then $Q[i] = H[j]$.

3.2. Analysis of Matching Method. The analysis of matching method is done in four different parts.

3.2.1. Consider a DNA Sequence and Their Related Changes

1 2 3 4 5 6 7 8 9 10 11 12. . . . . n
DNA CG G A A C T A A A C T C . . . . n_n
RNA CG G A A C U A A A C U C . . . . n_n
cDNA G C C T T G A T T T G A G . . . . n_n
cRNA GC C U U G A U U U G A G . . . . n_n,

where, n is the number of bases in the nucleotide sequence.

n_n is the nth (i.e., last) base (A/T/G/C) in host and query genome sequences, which consist of bases A, T, G, and C (note that T is replaced with U in the case of the RNA). This example is applicable both in host and query sequences, and n is the length of the sequence in both cases, but they are the same or do not depend on user.

3.2.2. Generating the Query Subsequence from Input Sequence. They broke the host and query sequence into user requirement subsequences length for easy implementation of Figure 1.

From Figure 1 pictorial representation, it is clear that for ith subsequence W_i (called colons): i is the starting position of the subsequence and $j = (i - 1) + L$ is end position of the subsequence, where L is the subsequence length (word size). For example, if word size is 4, then:

For

W_1 starting position $(i) = 1$ and (end position) $j = (1 - 1) + 4 = 4$,

W_2 starting position $(i) = 2$ and (end position) $j = (2 - 1) + 4 = 5$ and

W_3 starting position $(i) = 3$ and (end position) $j = (3 - 1) + 4 = 6$ and so on.

The clones with word size less than 3 (three) has no importance in matching context and hence we considered the clones with word size in the range: $3 \leq L \leq n$.

Therefore, ranges for i and j are as $3 \leq i \leq n - L + 1$ and $L + 1 \leq j \leq n$, respectively.

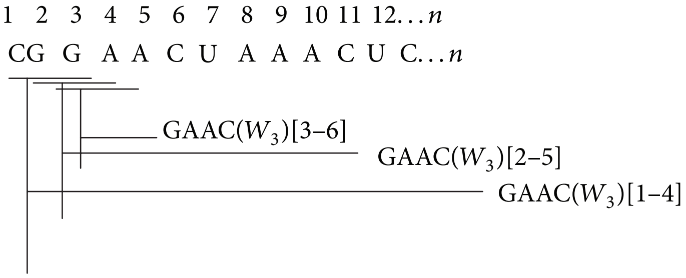


Figure 1

Table 1

Source sequence		Target sequence
DNA	⟶	DNA
RNA	⟶	RNA
cDNA	⟶	cDNA
cRNA	⟶	cRNA

The subsequence generation time, both in host and query sequences cases, at the end (subsequence length – 1) number of nucleotide base pair (a, t, g, and c) remains as it is. This is the reason why probability of infection decreases. To solve this problem, we have to find the result in reverse order.

The host sequence is defined by H and query sequence is defined by Q; each of the sequences must have the same or different lengths.

So, we could write
H = ATGCTAGCAGTAGACGATAGC.........n, $n > 0$ and T = TGCAGTAGCAGATGAC............m, $m > 0$, where n and m are the length of host and query sequences.

After subsequence division, they could get the result as follows.

So, they could rewrite $H[i] = H[1]H[2]\ldots\ldots\ldots H[n-L+1]$, $1 \leq i \leq n-L+1$ and $Q[j] = Q[1]\ Q[2]\ldots\ldots\ldots Q[m-L+1]$, $1 \leq j \leq m-L+1$.

If the subsequence length or word size is $L\,(3 < L \leq n-L+1)$.

If the number of subsequence is S, the total number of subsequences is generated in case that host sequence is $1 \leq S \leq n-L+1$ and case that query sequences is $1 \leq S \leq m-L+1$.

This subsequence method is required to reduce the complexity of the program execution.

3.2.3. Matching between Host and Query Sequence. Let us look for matches in between Host sequence and Query sequence in Table 1.

Here, host sequence is the virus sequence and Query sequence is the Tomato chloroplast, . . . and so forth, complete genome sequence of the Tomato plant and Root sequence.

16 possible matches may occur, and matches found are shown in the following:

DNA versus DNA

DNA versus RNA

DNA versus cDNA

DNA versus cRNA

RNA versus DNA

RNA versus RNA

RNA versus cDNA

RNA versus cRNA

cDNA versus DNA

cDNA versus RNA

cDNA versus cDNA

cDNA versus cRNA

cRNA versus DNA

cRNA versus RNA

cRNA versus cDNA

cRNA versus cRNA.

In these cases, the value of i is incremented by i = no. of unmatched character + no. of substring match × 3; similarly j is incremented by this same procedure.

Otherwise $Q[i] \neq H[j]$; that is, unmatched occurs, the value of i and j is incremented by one.

At the end, we could get the result as Table 2.

Host and Query sequence infections are calculated by |NBM|/||TL| where NBM is the total no of base pair match, which is equivalent to total number word match multiplied by word size, is divided by length of host sequence in case of virus infection, length of query sequence in case of plant infection.

3.2.4. Threshold Value. Proving this hypothesis, we have considered a threshold value, on this threshold value we can take the decision as described as follows.

(i) Infectivity "HIGH" means that the virus is highly infectious on target sequence; that is, chloroplast of the tomato plant is infected by PSTVd virus from head to tail. In this situation, the infection between the source (PSTVd) and the target sequence (tomato chloroplast) is very high.

(ii) Infectivity "NEGLIGIBLE" means that the virus is infected on target sequence; that is, chloroplast of the tomato plant is infected by PSTVd virus from head to tail are not infected. In this situation, the infection

Table 2

		$H[1]$ $H[5]H[6]$ ....$H[n-L+1]$
Source sequence	$S[i]$	: CGG C U AAAC.....................n
Target sequence	$T[i]$	: CG G A A C U A A A C U C.........m
		$T[1]$ $T[4]T[5]$ ....$T[m-L+1]$
Total word match = 3		

Table 3: Pictorial representation for showing the match region.

Position	Match position	Total base pair match	Gap	Highest match position without gap	Highest match position with gap
1st position	1–6 (1–3 and 4–6)	6	0		
2nd position	8–10	3	1		
3rd position	12–14	3	1		
4th position	17–22	6	2		
5th position	25–36	12	2	25–33	
6th position	38–39	3	1		25–39

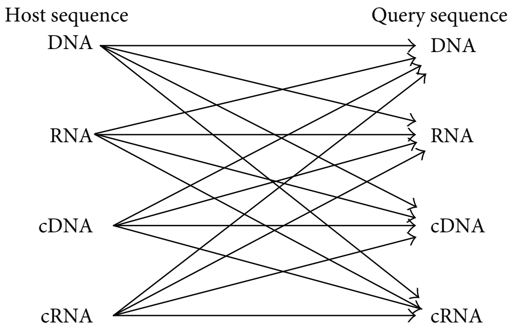


Figure 2: Matches between Host Sequence and Query Sequence.

between the source (PSTVd) and the target sequence (tomato chloroplast) is infected, but it is not harmful.

(iii) Infectivity "LOW" means the virus infection is found, but not so called infectious on target sequence; that is, chloroplast of the tomato plant is infected by PSTVd virus from head to tail are not infected. In this situation, the infection between the source (PSTVd) and the target sequence (tomato chloroplast) is noninfectious.

4. Experimental Data

4.1. Matches between Host Sequence and Query Sequence. This aspect is given in Figure 2.

4.2. Alignment Demo. The matter of alignment is shown in Figure 3.

4.3. Pictorial Representation Shows That Match Region. The pictorial representation of matched region is shown in Table 3 (word size 3).

Table 4: Highest matching word.

Words/colones	Repeat numbers
ATG	3
TTT	5
TAT	1
TGC	1

4.4. Highest Matching Word. The highest matched word is given in Table 4.

5. Project Spectrum

We have the following:

(i) A base program to detect the HCRs in a target sequence for a given viral sequence.

(ii) A method to locate the start and end positions of infection and isolate the infected regions.

(iii) A method to identify the longest infected region or the largest HCR.

(iv) An extension to allow all 4 possible transforms of the viral sequence (i.e., DNA, RNA, cDNA, and cRNA).

(v) An extension to allow scanning of all possible transforms of the normal plant (target) sequence, that is, DNA, RNA, cDNA, and cRNA. A total of 4×4 scan orientations.

(vi) An extension to identify successive regions of *Edit Distance* = 1.

(vii) An extension to detect and report all such extrapolated infection regions and locate the largest of them.

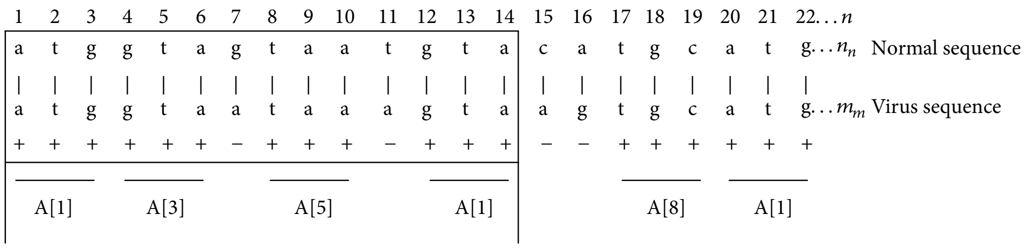

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22...n
a t g g t a g t a a t g t a c a t g c a t g...n_n Normal sequence
a t g g t a a t a a a g t a a g t g c a t g...m_m Virus sequence
+ + + + + + − + + + − + + + − − + + + + + +
A[1]
A[3]
A[5]
A[1]
A[8]
A[1]

Figure 3: Alignment demo.

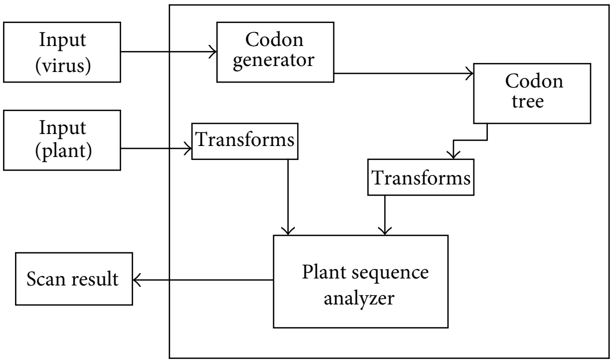


Figure 4: Architecture of process.

6. Architecture of Process

The required architecture for the whole process is shown in Figure 4.

6.1. Inputs

(i) The Inputs Taken are

(a) normal plant sequence:

(1) a steam of DNA bases in FASTA format, that is, a text file containing an DNA sequence.

(2) limitations: none.

(b) viral sequence:

(1) a steam of RNA bases in fasta format, that is, a text file containing an RNA sequence.

(2) limitations: size of file should be less than 400 Kbytes.

6.2. Codon Generator. Codon Generator is shown in Figure 5.

6.3. Codon Tree. The structure of codon tree is given in Figure 6.

6.4. Transforms. The process of transformation is shown in Figure 7.

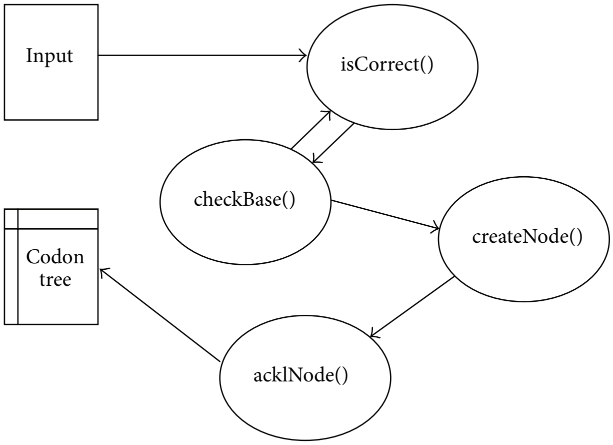


Figure 5: Codon generator.

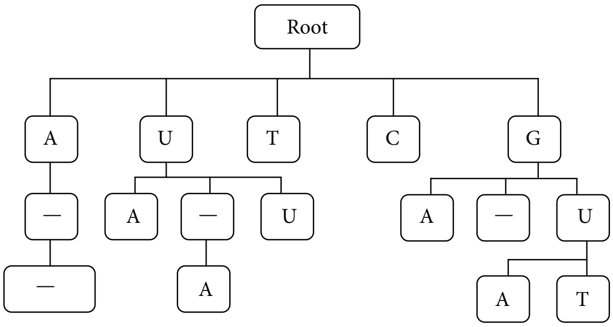


Figure 6: Codon tree.

6.5. Sequence Analyzer. The process of sequence analyzer is given in Figure 8.

7. Complexity

The algorithm uses an M-array tree to structure the input sequence and then allows the target to "pour through" the root and fit in place. Thus, the target sequence looks at a match, rather than the other way round. Here, $M = 5$ so the time complexity of the program is

$$\mathrm{O}(n_1 \log_M \mathrm{O}(n_1 \log_5 n_2) n_2)$$

$$\mathrm{O}\ (n_1 \log_5 n_2)$$

n_1: size of viral sequence

n_2: size of plant sequence.

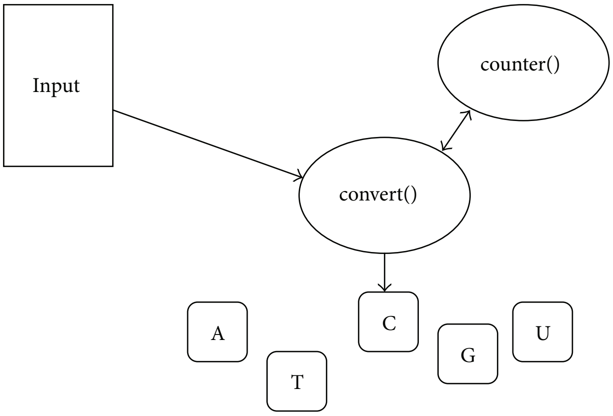


Figure 7: Process of transformation.

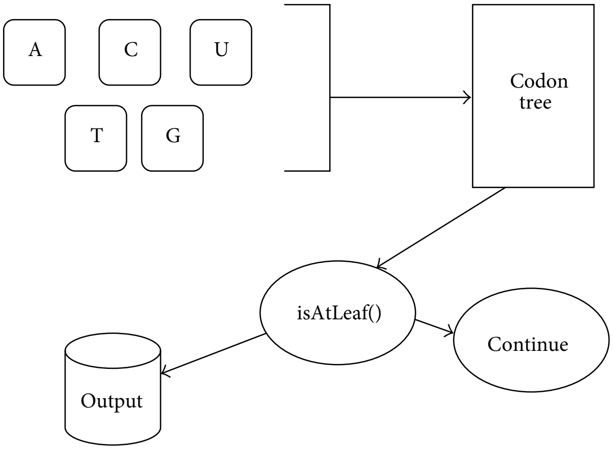


Figure 8: Sequence analyzer.

Table 5: Analysis of present algorithm.

Target input with fixed base sequence, 349 bytes	Time with strcmp()	Time with this Algorithm
200 KB	200 seconds	25 milliseconds
1 MB	7 minutes	456 milliseconds
1.5 MB	15 minutes	1-2 second (s)
>2 MB	The computer hanged	~15 seconds

8. Analysis

A comparison of a variant of the same program, using the strcmp() library function yielded the following timings. This is tabulated in Table 5.

9. Performance

The program was tested with real inputs and the time spent is tabulated in Table 6.

Table 6: Performance of viruses of different size.

Virus (in KB)	Plant (in KB)	Time taken
<400 bytes	<5	~0.5 milliseconds
500–1024 bytes	<5	~0.5 milliseconds
1–5	<100	~90 milliseconds
1–5	200–1024	~400 milliseconds
10–100	1024–5 MB	~1–4 seconds
10–100	5–7 MB	~5–10 seconds
100–300	~10	~15–20 seconds

10. Conclusion

This algorithm shows that virus and normal plant interaction was found only in between virus RNA with normal plant cDNA and RNA stand only. The virus and plant interaction was found only in normal in nature, no such other orientation is applicable. The colon size varies from 3 to 9. The lower the subsequence size, the higher the interaction rate. This algorithm also can apply on any type of virus and any type of normal plant genome sequences. In future, an attempt will be made to apply this software in real-life example such as Potato Spindle Tuber Viroid infected only chloroplast of the Tomato plant not in their root.

References

[1] T. O. Diener, *Viroids and Viroid Diseases*, Wiley, New York, NY, USA, 1979.

[2] H. J. Gross, H. Domdey, and C. Lossow, "Nucleotide sequence and secondary structure of potato spindle tuber viroid," *Nature*, vol. 273, no. 5659, pp. 203–208, 1978.

[3] H. J. Gross, G. Krupp, H. Domdey et al., "Nucleotide sequence and secondary structure of citrus exocortis and chrysanthemum stunt viroid," *European Journal of Biochemistry*, vol. 121, no. 2, pp. 249–257, 1982.

[4] H. J. Gross, U. Liebl, H. Alberty et al., "A severe and a mild potato spindle tuber viroid isolate differ in three nucleotide exchanges only," *Bioscience Reports*, vol. 1, no. 3, pp. 235–241, 1981.

[5] J. Haseloff and R. H. Symons, "Chrysantemum stunt viroid: primary sequence and secondary structure," *Nucleic Acids Research*, vol. 9, no. 12, pp. 2741–2752, 1981.

[6] J. E. Visvader, A. R. Gould, G. E. Bruening, and R. H. Symons, "Citrus exocortis viroid: nucleotide sequence and secondary structure of an Australian isolate," *FEBS Letters*, vol. 137, no. 2, pp. 288–292, 1982.

[7] R. H. Symons, "Avocado sunblotch viroid: primary sequence and proposed secondary structure," *Nucleic Acids Research*, vol. 9, no. 23, pp. 6527–6537, 1981.

[8] J. Haseloff, N. A. Mohamed, and R. H. Symons, "Viroid RNAs of cadang-cadang disease of coconuts," *Nature*, vol. 299, no. 5881, pp. 316–321, 1982.

[9] P. Van Wezenbeek, P. Vos, J. van Boom, and van Kammen, "A unique mechanism regulating gene expression: translational inhibition by a complementary RNA transcript (micRNA)," *Nucleic Acids Research*, vol. 10, pp. 794–797, 1982.

[10] H. J. Gross and D. Riesner, "Viroids: a class of subviral pathogens," *Angewandte Chemie*, vol. 19, no. 4, pp. 231–243, 1980.

[11] T. O. Diener, “Are viroids escaped introns?” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 78, no. 8 I, pp. 5014–5015, 1981.

[12] E. Dickson, “A model for the involvement of viroids in RNA splicing,” *Virology*, vol. 115, no. 1, pp. 216–221, 1981.

Hindawi Publishing Corporation
BioMed Research International
Volume 2013, Article ID 634832, 11 pages
http://dx.doi.org/10.1155/2013/634832

Research Article

HyDEn: A Hybrid Steganocryptographic Approach for Data Encryption Using Randomized Error-Correcting DNA Codes

Dan Tulpan,[1,2] Chaouki Regoui,[1] Guillaume Durand,[1,3] Luc Belliveau,[1] and Serge Léger[1]

[1] *National Research Council Canada, 100 des Aboiteaux Street, Moncton, NB, Canada E1A 7R1*
[2] *Department of Biology, Université de Moncton, Moncton, NB, Canada E1A 3E9*
[3] *Department of Computer Science, Université de Moncton, Moncton, NB, Canada E1A 3E9*

Correspondence should be addressed to Dan Tulpan; dan.tulpan@nrc-cnrc.gc.ca

Received 30 April 2013; Accepted 29 June 2013

Academic Editor: Tai-hoon Kim

This paper presents a novel hybrid DNA encryption (HyDEn) approach that uses randomized assignments of unique error-correcting DNA Hamming code words for single characters in the extended ASCII set. *HyDEn* relies on custom-built quaternary codes and a private key used in the randomized assignment of code words and the cyclic permutations applied on the encoded message. Along with its ability to detect and correct errors, *HyDEn* equals or outperforms existing cryptographic methods and represents a promising *in silico* DNA steganographic approach.

1. Introduction

The deluge of counterfeited goods flooding the world markets today generates a high demand for novel cryptographic and steganographic approaches that will better protect information and branded products and ensure their authenticity. Positioned at the confluence of mathematics, biology, informatics, chemistry, and physics, cryptography and steganography represent the ultimate means for information protection.

1.1. Cryptography. Cryptography is generally defined as the practice and study of techniques for secure communication performed over unsecured channels. There are two major operations involved in secure communication, namely, the encryption and decryption of a message. The purpose of encryption is to modify the information, such that only an authorized party is capable of decoding it. Both, encryption and decryption, require a key, which is needed by the authorized parties, and it is assumed to be kept secret. To date, only one encryption approach was mathematically proven to be secure and virtually unbreakable: the one-time pad [1]. Nevertheless, its practicality is hampered by the necessity of a random key, which must be at least as long as the message itself. For all other cryptographic approaches, there is a theoretical possibility of breaking them, although the time required to do so might be very long, thus making the approaches fairly secure. Examples of such cryptographic approaches include the data encryption standard (DES) [2], the advanced encryption standard (AES) [3], the Rivest-Shamir-Adleman (RSA) method [4], and the Pretty Good Privacy (PGP) [5] method.

1.2. Steganography. Steganography is the science of concealing information within different types of media, such that only the sender and the receiver are aware of its exact location. Unlike cryptography, where only the message is protected, steganography protects both the message and the communicating parties. With origins deeply rooted in ancient Greece, where messages were recorded as texts or tattoos and then hidden on wax tablets and skins, steganography was used relentlessly over the centuries under various ingenious forms such as invisible inks [6], postal stamps [7], knitted clothes [8], microdots [9], modified images [10], executable files [11], and DNA sequences embedded in various materials [12, 13].

1.3. Error-Correcting DNA Codes. Error-correcting codes consist of sets of symbols defined over a finite alphabet, such that if any code word is altered in t positions we can detect and correct the error based on knowledge of the remaining code words.

For example, assume a given binary code W consisting of two code words $w_1 = 000$ and $w_2 = 111$ each of length 3. A 1-bit error occurring in any of the two code words (e.g., w_2) will produce a modified code word; let us say $w_2' = 101$. By comparing the modified code word w_2' with both code words from W, we notice that it differs in only one bit from w_2 (middle bit), while it differs in two bits compared with w_1 (flanking bits). Thus, we can quickly identify the exact location of the error and correct it based on w_2's closest proximity to code word w_2.

1.3.1. Hamming Codes. One instance of simple and efficient error-correcting codes are Hamming codes [14], where each pair of code words differs in at least d bits. We denote by $A_4(n, d)$ the size of a quaternary code where all pairs of code words of length n differ in at least d positions. The number of bits/positions in which two code words differ is also known as the Hamming distance. For certain combinations of n and d, the exact size of quaternary codes are unknown and thus lower and upper bounds were derived to provide approximations. The text by MacWilliams and Sloane [15] provides a succinct introduction to the topic.

While Hamming codes were originally designed using a $\{0, 1\}$ alphabet with the purpose of sending binary information over noisy channels, the increased need for storing and retrieving information with synthetic DNA strands used as chemical bar codes, or as biological tags for DNA computing applications, facilitated the advent of Hamming codes defined over quaternary alphabets, such as the DNA alphabet $\{A, C, T, G\}$.

1.3.2. DNA Codes. A single-stranded DNA molecule is a long, unbranched polymer composed of only four types of subunits linked together by chemical bonds and attached to a sugar-phosphate chain like four kinds of beads strung on a necklace. These subunits are the deoxyribonucleotides containing the bases: adenine (A), cytosine (C), guanine (G), and thymine (T).

Conceptually equivalent to a digital signal, DNA sequences are naturally and synthetically used for information encoding in living organisms and biotechnological and steganographic applications. Given the data encoding capacity of DNA and the fact that traditional data encoding techniques using binary sequences are fortified against communication errors, quaternary codes using the DNA alphabet $\{A, C, T, G\}$ were proposed and continuously developed over the past decades.

The design of error-correcting DNA codes of fixed length n that satisfy various combinations of constraints such as having a minimum pairwise Hamming distance ($d_{\min}$) is a hard computational problem, whose complexity is still unknown today. Over the past two decades, a large number of publications have proposed intricate code design techniques [16–18] based on their state-of-the-art algorithms such as stochastic local search, genetic algorithms, and pure mathematical constructions. Most of these approaches lead also to the continuous improvement of upper and lower bounds for DNA codes [19–21].

Assuming that a DNA code C with k code words of length n is given and that each pair of distinct code words w_i and w_j obeys the condition that, for all pairs (w_i, w_j) with $i, j \in N$, $i \neq j$,

$$\text{Hamming Distance}\left(w_i, w_j\right) \geq d, \quad (1)$$

then C can detect $\lfloor d/2 \rfloor$ errors and can correct $\lfloor (d-1)/2 \rfloor$ errors.

1.4. Related Work. Over the past decade, complex algorithms have been devised to encode information using DNA sequences. Examples of such algorithms include the DNA triplet-based approach described by Clelland et al. [9], which extends the principle of using microdots to hide information developed during the Second World War. An extension of Clelland et al.'s work was presented by Leier et al. [22], and it consisted of encoding zeros and ones using short DNA sequences with sticky ends, which can bind together forming longer sequences. The encrypted messages include a mixture of coding and noncoding DNA sequences, and the decryption can be performed only by someone who has access to the correct primer sequences. A primer is a short DNA sequence that serves as a starting point for DNA synthesis. A similar approach based on DNA tiling was proposed by Hirabayashi et al. [23] who designed true random one-time pads using a DNA cryptosystem. The true randomness is conferred by molecular computations using hybridization of DNA sequences encoding 4 types of cipher tiles.

Gehani et al. [24] extended the one-time pad approach to perform operations on DNA sequence pairs, representing plain and cipher texts. Originally, the one-time pad approach was designed to perform XOR operations on binary codes. The message encoded with DNA pairs can be retrieved and decoded using specific DNA polymerases. Arita and Ohashi [25] developed a steganographic algorithm based on the redundant codon table (see Table 1). A codon consists of 3 consecutive nucleotides, and while it is possible to have 64 (4^3) different codons, only 20 of them encode distinct amino acids, with the rest being redundant. Their algorithm encoded each letter in the English alphabet using binary codes of length 5, with each bit being encoded by a codon. They added an additional parity bit to each letter encoding to keep the number of bits in each bit-pattern odd and thus used for error-detection purposes. The decoding could be achieved only by someone who knows the original codon sequence.

Following a different approach, Wong et al. [27] developed a DNA steganography method that encodes information in living organisms. The information is encoded with the aid of unique DNA sequences that do not exist in the particular genomes where they will be embedded, thus assuring the success of the identification stage. For this approach to succeed, the embedded foreign DNA must be replicated by the host organism together with their genomic DNA. The extraction of the information is achieved using a

Table 1: The redundant DNA codon table.

Amino acid	DNA codons					
Alanine	GCT	GCC	GCA	GCG		
Arginine	CGT	CGC	CGA	CGG	AGA	AGG
Asparagine	AAT	AAC				
Aspartic acid	GAT	GAC				
Cysteine	TGT	TGC				
Glutamic acid	GAA	GAG				
Glutamine	CAA	CAG				
Glycine	GGT	GGC	GGA	GGG		
Histidine	CAT	CAC				
Isoleucine	ATT	ATC	ATA			
Leucine	CTT	CTC	CTA	CTG	TTA	TTG
Lysine	AAA	AAG				
Methionine	ATG					
Phenylalanine	TTT	TTC				
Proline	CCT	CCC	CCA	CCG		
Serine	TCT	TCC	TCA	TCG	AGC	AGT
Threonine	ACT	ACC	ACA	ACG		
Tryptophan	TGG					
Tyrosine	TAT	TAC				
Valine	GTT	GTC	GTA	GTG		
Start (CI)	ATG					
Stop (CT)	TAA	TAG	TGA			

standard laboratory technique called the polymerase chain reaction (PCR) [28].

The DNA-Crypt approach proposed by Heider and Barnekow [29] combines and extends the steganographic and cryptographic methodologies proposed by Wong et al. [27] and Arita and Ohashi [25]. DNA-Crypt encodes information using a substitution cipher and two types of error-correcting codes, namely, Hamming [14] and WDH [30]. DNA-Crypt incorporates a fuzzy controller and powerful cryptographic algorithms such as one-time pad, AES, Blowfish [31], and RSA. Shiu et al. [32] introduced 3 data hiding methods based on properties of DNA sequences, namely, the insertion method, the complementary pair method, and the substitution method. All three methods provide distinct means to incorporate secret messages within existing DNA sequences pulled from public databases. The known DNA sequence acts as a private key, and it can be identified only by the sender and the receiver.

A hybrid approach built on the substitution method described in Shiu et al. [32] that combines cryptography and DNA steganography was proposed by Torkaman et al. [33]. Their approach uses reference DNA sequences from the European Bioinformatics Institute (EBI) Database, which contains roughly 163 million entries. The encoding of information is achieved using 6 association rules.

Here, we present the hybrid DNA encryption (HyDEn) approach, which combines the advantages conferred by cryptography and steganography into a unique symmetric cryptosystem. The system uses a unique private numeric key to scramble the assignment of DNA code words from a predesigned set to the extended ASCII characters and then apply a cyclic permutation on the encrypted message. The combination of key uniqueness, the randomization of code word assignments, the undisclosed code word length, and the final cyclic permutation of the encrypted message confer additional strength to the proposed approach. The information encrypted with HyDEn can be safely communicated between senders and receivers via dedicated and inconspicuous publicly accessible channels, such as bioinformatics discussion groups and DNA sequence databases.

2. HyDEn: The Hybrid DNA Encryption Approach

Deeply rooted in the ways nature encodes information using nucleic acids, DNA stegano-cryptography uses short DNA sequences to encrypt and hide messages, thus protecting their content. The hybrid DNA encryption (HyDEn) approach presented here includes a novel *in silico* cryptosystem that uses DNA error-correcting Hamming codes and disguises encrypted messages as long DNA sequences conveniently placed on host bioinformatics resources.

Following next is a stepwise description of the HyDEn cryptosystem.

Input. The message is defined over an alphabet Ω, private key pk.

Encryption Algorithm

Step 1. Select an error-correcting DNA code with $|\Omega|$ n-ary code words obtained with one of the state-of-the-art code design techniques described in Aboluion et al. [16], Gaborit and King [19], Tulpan and Hoos [26], and Tulpan et al. [18]. Here, n represents the number of characters in a DNA code word. An example of a DNA code with $n = 8$ and $d = 3$ is given in Table 2.

Step 2. Using the key pk provided as input, perform a random shuffling of the n-ary DNA code words that will be associated to each character from Ω.

Step 3. Encrypt the message using the random assignment of DNA code words obtained in Step 2.

Step 4. Perform a circular rotation (mod$|\Omega|$) to the right of the characters in the message with exactly pk positions.

Output. The encrypted message m.

Step 1 provides the means of encoding a message using a code defined over a quaternary alphabet. The code will be able to identify and correct errors that can occur during the message transmission stage. Step 2 will generate a unique code word assignment based on the key pk. If all pk keys are unique, then the assignment will be equivalent to a one-time pad system. In the eventuality that code word length (n) is found, Step 4 is used to lower the chances of a successful frequency analysis based on well-established tests such as the Friedman test [34] and the Kasiski test [35].

Table 2: A sample DNA $A_4(8,3)$ Hamming code consisting of 256 code words. Each code word can be associated with an extended ASCII character and used for encoding text messages. The code was obtained with the DNA word design algorithm described in Tulpan and Hoos [26].

A set with 256 code words							
AAAAAAGA	ACTACACT	ATGGAGTT	CCCTTCGA	CTGGTAGT	GGAAAGGT	GTTGTATT	TCGTGTTA
AAAAGAAG	ACTACCTA	ATGGGAAG	CCGATTTC	CTGGTTCG	GGATGACA	TAACATAC	TCTCCGAG
AAAATGTT	ACTCTCAG	ATGTAAGT	CCGCGCAT	CTTCGGTG	GGCCAAGT	TAACCATA	TCTCCTTA
AAACCTGC	ACTGGAGT	ATTCATAC	CCGGCGCG	CTTGACAT	GGCCGACG	TAACGAGG	TCTGCGCA
AAACTCAC	ACTTCCGC	ATTCTGCG	CCGTAGCC	CTTGCATG	GGCCTGGA	TAAGAGCA	TCTGGCTC
AAAGATCG	ACTTGCAT	ATTTAATC	CCGTTCAG	CTTTCCAC	GGCGTGCC	TAAGTTGA	TCTGTTAC
AAATGTGG	ACTTTGGG	ATTTCAGA	CCTACCGG	GAATCATC	GGCTGCAT	TAATAGGC	TGAAAATA
AAATTGAG	AGACCCTA	CAAATACG	CCTTCTGT	GACAGCGT	GGGCATAC	TAATGGAA	TGACTCAT
AACAGCTG	AGACTTAA	CAAATCTA	CCTTGTCG	GACCAGCT	GGGCTTGG	TAATTACT	TGAGCATC
AACCTAGC	AGAGCGGT	CAATATGA	CCTTTGAC	GACCGTTA	GGGGCCCA	TACGCAAA	TGAGGGTT
AACGCGTT	AGAGTAAT	CAATTCGC	CGAACGCT	GACGGTAT	GGGGGTTC	TACTTGGG	TGATATAT
AACGGTGA	AGATCTTG	CACCTAAT	CGACCTTT	GAGAATTA	GGTAATGG	TAGACTGA	TGATTCGG
AACTACGT	AGATGGCT	CACTCGAA	CGAGAAAC	GAGAGAGC	GGTACGTA	TAGAGTAC	TGCATAAG
AACTCATA	AGCCAGCA	CAGACAGG	CGAGCGTA	GAGAGTCG	GGTATGCG	TAGGAGTG	TGGGGCGC
AAGAAACT	AGCTCGGG	CAGCAACG	CGAGCTCG	GAGTTGTT	GGTTTAGT	TAGTAACC	TGGTTTTT
AAGATAAC	AGGACTGT	CAGCCGGC	CGAGTCTT	GATACCCC	GGTTTCCC	TAGTCCGG	TGTCAGAT
AAGCACGC	AGGATGAG	CAGGTCGA	CGATGTAC	GATATTGC	GTAACGCG	TATAAATG	TGTGCAAT
AAGGTTGT	AGGCCCAT	CAGTGATC	CGCCACGA	GATCATAT	GTACTACG	TATATGGT	TGTGTTGG
AATAGTCT	AGGTACTT	CATCGAGC	CGCCTCCC	GATCCCAG	GTAGATCA	TATGTGAA	TTAAGCCG
AATCGTTC	AGGTAGGC	CATCTTTG	CGGAAGTA	GATGACTA	GTAGTCGT	TCAAACGC	TTAATTTA
AATGCGGG	AGGTGTCC	CATGCTTA	CGGTAACA	GATTGTTG	GTCATATG	TCAAAGTG	TTAGCTGT
AATGTGCT	AGTCGAAG	CATGGGGA	CGGTGTTG	GATTTACG	GTCCGAAT	TCAAGAAC	TTAGTCCA
AATTGGTT	AGTCGGGA	CCACCGCC	CGTCACAC	GCAGGTCG	GTCCTTAA	TCACAAGA	TTCAAGAC
ACACTAGT	AGTGCCGA	CCAGATGC	CGTTAGCT	GCATTCTT	GTCGCAAG	TCAGTGCC	TTCCGCAC
ACACTTCC	ATATGCCC	CCAGTATC	CTAACTCC	GCATTTCA	GTCTCCAA	TCATCTTC	TTCGAATA
ACAGCTTA	ATCACAAA	CCAGTGGA	CTAGACGG	GCCGAATT	GTGGAGAA	TCCGAGGC	TTGCGTTC
ACATCGAA	ATCACCGG	CCATGACC	CTAGAGCC	GCCGCGGT	GTGGCCAT	TCCGCCGA	TTGGGGTA
ACCGGATC	ATCCCTGA	CCATGCAA	CTATTACA	GCGAATGT	GTGTCGGT	TCCTGAAG	TTGTCTTG
ACCTCAAC	ATCGTAGG	CCCCTACG	CTATTGTT	GCGACATT	GTTATCAC	TCGATGCG	TTTACAGC
ACGCATTT	ATCTCTTC	CCCGGAGA	CTCCCAGT	GCGGGTAA	GTTCACTG	TCGGAACA	TTTCCACG
ACGCTATG	ATCTTCAC	CCCGGGAG	CTCCGGCC	GCTGAGTG	GTTCCAAC	TCGTAGAG	TTTCGTAG
ACGTCGTC	ATGACGTG	CCCTAGTT	CTCGCGGC	GCTGTCCG	GTTGCTCT	TCGTCCAT	TTTGTGTG

The message decryption step will use the same unique key to perform the reverse circular permutation on the encrypted message and find the correct code words assignment, which will reveal the original message.

The flowcharts for message encryption and decryption with HyDEn are summarized in Figure 1.

3. Example of Message Encryption and Decryption Using HyDEn

To better understand how the HyDEn approach works, let us assume that Alice would like to transmit the message "ATTACK AT DAWN" to Bob. They have established before hand to use the secret key "5". The message uses only 8 distinct ASCII characters, namely, "space," "A," "C", "D," "K," "N," "T" and "W." Based on the unique key used by Alice and Bob, and applying Steps 1 and 2 of our approach, a unique assignment of DNA code words of length 8 is associated to each of the 8 characters, as shown in Table 3.

Table 3: A sample assignment of code words to ASCII characters.

DNA code word	ASCII character
AAAAAAGA	→ space
ACTACACT	→ A
ATGGAGTT	→ C
CCCTTCGA	→ D
CTGGTAGT	→ K
GGAAAGGT	→ N
GTTGTATT	→ T
TCGTGTTA	→ W

Using this assignment, the encrypted message resulting after Step 3 is the following:

ACTACACTGTTGTATT**GTTGTATT**ACTACACT
ATGGAGTTCTGGTAGT**AAAAAAGA**ACTACACT
GTTGTATTAAAAAAGA**CCCTTCGA**ACTACACT
TCGTGTTAGGAAAGGT

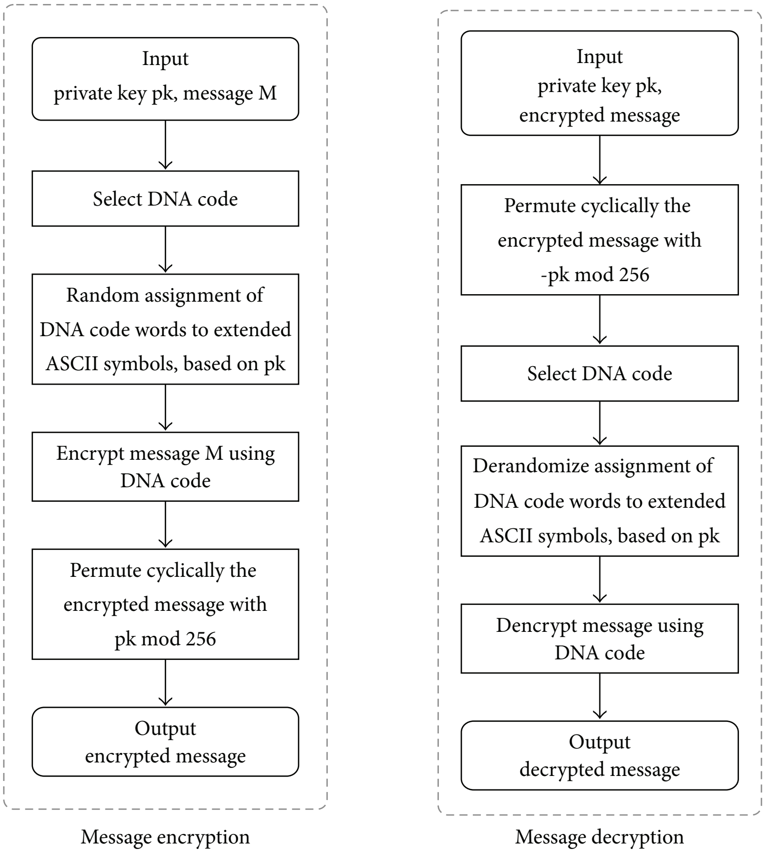

Input
private key pk, message M
Select DNA code
Random assignment of DNA code words to extended ASCII symbols, based on pk
Encrypt message M using DNA code
Permute cyclically the encrypted message with pk mod 256
Output
encrypted message
Message encryption
Input
private key pk,
encrypted message
Permute cyclically the encrypted message with -pk mod 256
Select DNA code
Derandomize assignment of DNA code words to extended ASCII symbols, based on pk
Dencrypt message using DNA code
Output
decrypted message
Message decryption

Figure 1: Flowcharts for message encryption and decryption with HyDEn.

To better visualize the encryption process, every second code word was bold faced. The encrypted message is then permuted cyclically five positions to the right, thus obtaining the following sequence of DNA bases:

AAGGT**ACTACACT**GTTGTATT**GTTGTATT**
ACTACACT**ATGGAGTT**CTGGTAGT**AAAAAAGA**
ACTACACT**GTTGTATT**AAAAAAGACCCTTCGA
ACTACACT**TCGTGTTA**GGA

Ideally, the key (mod 256) must be different from a multiple of the code word length (n); otherwise, the permutation will shift the encrypted message exactly n letters to the right (or to the left) and will not have the desired effect.

4. Comparison Parameters

To facilitate the comparison between our approach and related encryption methodologies, we use a combination of performance parameters including the ones introduced by Shiu et al. [32], namely, capacity, payload, *bpn*, and the cracking probability or the probability of a successful brute-force attack P_{bf}.

The capacity (C) is defined as the total length of a reference sequence that encodes or includes the encrypted message. The payload (P) is the remaining length of the new sequence after subtracting the reference DNA sequence. The *bpn* represents the number of hidden bits per character. The previous parameters utilize the following notations: n is the length of a DNA sequence, m is the message that will be encrypted, and $|m|$ is its length.

5. Results and Discussion

We analyze the robustness of HyDEn by estimating the probability of success for a brute-force attack, and we provide a comparative assessment between our cryptosystem and other cryptographic techniques with performance characteristics described in the literature. The comparison relies on a set of parameters introduced in Section 4. We further investigate HyDEn's strengths and weaknesses, and we provide insights into potential improvements that will augment its performance.

5.1. Robustness. Calculations of the strength of encryption against brute-force attacks are typically the worst case scenarios thus, the probability of success for a brute-force attack against the proposed cryptosystem (HyDEn) is captured

$$P_{bf} = \frac{1}{n} \cdot \frac{1}{|\Omega|!} \cdot \frac{1}{|\Omega|}, \qquad (2)$$

where n is the length of a DNA code word and $|\Omega|$ is the number of characters in alphabet Ω.

Table 4: Comparison between *HyDEn* and other encryption methods. n is the length of a DNA sequence, $|m|$ is the length of the original message, $|\Omega|$ is the size of the DNA code, and k is a method-specific parameter that represents the length of the longest complementary pairs in the reference DNA sequence.

Method	C	P
HyDEn	n	0
Insertion [32]	$n + \frac{\lvert m\rvert}{n}$	$\frac{n}{2}$
Complementary pair [32]	$n + \lvert m\rvert \cdot (k + 3.5)$	$\lvert m\rvert \cdot (k + 3.5)$
Substitution [32, 33]	n	0
Method	bpn	P_{bf}
HyDEn	$\frac{\lvert m\rvert}{n}$	$\frac{1}{n} \cdot \frac{1}{\lvert\Omega\rvert!} \cdot \frac{1}{\lvert\Omega\rvert}$ (e.g., $\frac{1}{2^{11} \cdot e^{1163.6}}$)
Insertion [32]	$\frac{\lvert m\rvert}{n + \lvert m\rvert/2}$	$\frac{1}{1.63 \cdot 10^8} \cdot \frac{1}{n-1} \cdot \frac{1}{2^{\lvert m\rvert} - 1} \cdot \frac{1}{2^n - 1} \cdot \frac{1}{24}$
Complementary pair [32]	$\frac{\lvert m\rvert}{n + \lvert m\rvert \cdot (k + 3.5)}$	$\frac{1}{1.63 \cdot 10^8} \cdot \frac{1}{24^2}$
Substitution [32, 33]	$\frac{\lvert m\rvert}{n}$	$\frac{1}{1.63 \cdot 10^8} \cdot \frac{1}{6}$ or 3^n

Assuming that Ω is the extended ASCII character set, then $|\Omega| = 256$ and (2) becomes

$$P_{bf} = \frac{1}{n} \cdot \frac{1}{256!} \cdot \frac{1}{256}. \tag{3}$$

Using the Stirling approximation [36] for factorials, $\ln(k!) \approx k \cdot \ln(k) - k$, for all $k \in \mathbb{R}$, and DNA code word length $n = 8$, we obtain

$$P_{bf} \approx \frac{1}{2^{11} \cdot e^{1163.6}}. \tag{4}$$

The first term in (2) comes from the fact that n is unknown to the attacker; thus, a successful attacker must first guess the length of the used code words, which would be 8 in the sample $A_4(8,4)$ DNA code from Table 2. The second term of the equation describes the probability of finding the correct code assignment for the extended ASCII character set. We also assume that the attacker already knows what character set is encoded by the DNA code. The last term of the equation is given by the probability of finding the correct cyclic permutation applied to the encrypted message. Without knowing the correct permutation, the attempt of identifying the correct code word assignment is prone to failure.

5.2. Comparison with Other DNA Cryptographic Strategies. Using the parameter estimations described in Section 4, we compare HyDEn with other encryption approaches described in Shiu et al. [32].

Table 4 presents comparative results between HyDEn and other cryptographic methods. The methods are compared based on their capacity (C), payload (P), the number of hidden bits per character (bpn), and the probability of success for a brute-force attack (P_{bf}).

Based on the probability of success for a brute-force attack (P_{bf}), HyDEn and the insertion method are the most secure, while the substitution method seems to be the least secure. Nevertheless, the best capacity (C), payload (P), and bpn correspond to *HyDEn* and the Substitution method, while the insertion method ranks second and the complementary pair third.

The result expressed in (4) can be also directly compared with the result reported by Torkaman et al. [33] on page 233 in their paper. Their result states that the probability of recovering via a brute-force technique an original message hidden within a sequence database with other 163 million sequences is equal to $(1/(1.63 \times 10^8)) \times (1/6)$. Using simple numerical inequality manipulations, we show that our technique confers higher protection against brute-force attacks compared with the method proposed by Torkaman et al.:

$$\begin{aligned}
&\frac{1}{2^{11} \times e^{1163.6}} \\
&\quad < \frac{1}{2^{11} \times 2^{1163.6}} < \frac{1}{2^{11} \times 2^{1163}} = \frac{1}{2^{1174}} \\
&\quad \ll \frac{1}{2^{32}} = \frac{1}{2^4 \times 2^{28}} = \frac{1}{2 \times 2^3 \times 2^{28}} < \frac{1}{2 \times 6 \times 2^{28}} \\
&\quad < \frac{1}{2 \times 6 \times 10^8} < \frac{1}{1.63 \times 10^8} \times \frac{1}{6}.
\end{aligned} \tag{5}$$

Thus, P_{bf} (HyDEn) $\ll$ P_{bf} (substitution: Torkaman et al. [33]).

5.3. HyDEn's Strengths, Weaknesses, and Potential Extensions. Compared with the existing DNA-based cryptographic and steganographic methods, HyDEn has one of the lowest probabilities of success for brute-force attacks. HyDEn includes mechanisms such as cyclic permutations and randomized assignments of code words to protect against various types of frequency analysis such as the Kasiski and Friedman tests along with error detection and correction capabilities conferred by DNA Hamming codes. One of the drawbacks of using many-to-one character encoding schemes is the increase in size of the encrypted message, which could

Table 5: A sample DNA $A_4(8,3)$ Hamming code consisting of 1024 code words. Four distinct code words can be associated with one extended ASCII character and used for encoding text messages. The code was obtained with the DNA word design algorithm described in Tulpan and Hoos [26].

A set with 1024 code words							
AAAAAAAG	AAAAAGGA	AAAACTCC	AAAAGCAC	AAACAATA	AAACAGCT	AAACCGTC	AAACGAGG
AAACGCCA	AAACGTAT	AAAGATGG	AAAGTCTT	AAAGTGGC	AAATATAA	AAATCTTT	AAATGGCG
AAATTGAT	AACACAAA	AACAGACC	AACAGCTA	AACCAGGG	AACCATCA	AACCCTAC	AACCTCGA
AACCTGTT	AACGACCG	AACGCATC	AACGGGGA	AACGTTAA	AACTCGTG	AACTGTGC	AAGAAGAC
AAGACTAG	AAGAGGTG	AAGATACA	AAGATGGT	AAGATTTC	AAGCGTGA	AAGCTCAT	AAGCTGCC
AAGGACGC	AAGGCCTG	AAGGCGCA	AAGGGATA	AAGGTAAG	AAGTAATT	AAGTGCAG	AATAATTA
AATACACG	AATACCTT	AATCAACC	AATCCGGA	AATCGCTC	AATGGAGC	AATGGCAA	AATGGTTG
AATGTGCT	AATTACTG	AATTAGCA	AATTCAAC	AATTGGGT	AATTTAGG	AATTTTCC	ACAAAGTC
ACAACCCG	ACAAGAGT	ACAATTTT	ACACATTG	ACACCAAT	ACACCTCA	ACACTACG	ACACTGTA
ACAGCTGC	ACAGGGCC	ACAGGTAG	ACATAACT	ACATACGA	ACATCATG	ACATTCAC	ACCAACCA
ACCACTGA	ACCAGTTG	ACCATGAC	ACCCAATT	ACCCCCGG	ACCCGAAG	ACCCGGCT	ACCGACAC
ACCGAGTA	ACCGCGAT	ACCGTAGC	ACCTATCG	ACCTGACA	ACCTGCTC	ACCTTTAT	ACGAATGG
ACGACGCC	ACGAGGGA	ACGATCAG	ACGCAACA	ACGCCCAA	ACGCCGTT	ACGCGTAC	ACGCTGGG
ACGGCCGT	ACGGTTCT	ACGTAGAA	ACGTTAGT	ACTAAATG	ACTACGAA	ACTAGCCC	ACTATCTA
ACTATTGC	ACTCATGT	ACTGCACC	ACTGCTTT	ACTGGTCA	ACTGTCAT	ACTTATTC	ACTTCCAG
ACTTCGCT	ACTTGTGG	ACTTTAAA	ACTTTGTG	AGAACCAT	AGAATCTC	AGAATGCG	AGACAAAC
AGACGCGT	AGACGTTA	AGACTCAA	AGAGAGCA	AGAGCACT	AGAGCCGA	AGAGTATA	AGAGTTCC
AGATCGAC	AGATGAGC	AGCAACGT	AGCAATAA	AGCACTTC	AGCAGGCA	AGCATAAT	AGCCAGAT
AGCCCACA	AGCCGTGG	AGCGCCCC	AGCGGGAC	AGCGTCAG	AGCTAAGA	AGCTATTT	AGCTCAAG
AGCTGCAA	AGCTTGGT	AGGAACTG	AGGACATT	AGGAGTGC	AGGATGTA	AGGCAAGG	AGGCATCC
AGGCCTTG	AGGGACAT	AGGGCAAA	AGGGCGGC	AGGGGACC	AGGGGGTT	AGGTAGTC	AGGTCGCG
AGGTCTGT	AGGTGTCA	AGGTTCCC	AGGTTTAG	AGTAACAC	AGTAGGTC	AGTATCCT	AGTCAGTG
AGTCCAGT	AGTCCGCC	AGTGATCG	AGTGCTAC	AGTGGAAT	AGTGGCGG	AGTGTGGA	AGTTCCTA
AGTTGACG	AGTTTATC	ATAAACTT	ATAACATA	ATAATTCA	ATACCCAG	ATACGGTT	ATACTTGC
ATAGAAGT	ATAGCGTG	ATAGGTTC	ATAGTGAA	ATATAGGC	ATATCCTC	ATATCTGG	ATATGCCT
ATCAAGTG	ATCACGGC	ATCAGCCG	ATCAGTAC	ATCATTGT	ATCCAGCC	ATCCATAG	ATCCCCCT
ATCCCTTA	ATCCTAAA	ATCGAACA	ATCGATGC	ATCGCAGG	ATCGTGTC	ATCGTTCG	ATCTAATC
ATCTACAT	ATCTCTCC	ATCTGGAG	ATCTTCGC	ATCTTGCA	ATGAAACT	ATGAGATC	ATGAGCGT
ATGCATTT	ATGCCAAC	ATGCGGAA	ATGCGTCG	ATGGACTA	ATGGAGGG	ATGGGCAC	ATGGTAGA
ATGTACCG	ATGTCGGA	ATGTGAGG	ATTAAAAA	ATTAAGGT	ATTACCCA	ATTAGTCT	ATTATTAG
ATTCCATG	ATTCCCGC	ATTCGACA	ATTGACCT	ATTGAGAC	ATTGCTGA	ATTGGGCG	ATTGTATT
ATTTCTAT	ATTTGCGA	CAAAATCA	CAAACTGG	CAAAGGAA	CAAATCCC	CAACACTC	CAACCCAT
CAACGGCC	CAAGAATT	CAAGCCTA	CAAGCTCT	CAAGGCCG	CAAGGGGT	CAATACAG	CAATATGT
CAATGTTG	CAATTGCA	CACAAAAC	CACAAGTA	CACACCCG	CACATTCT	CACCAACG	CACCGCGG
CACCTTTC	CACGCAGT	CACGCTTG	CACTACGC	CACTAGAT	CACTCTCA	CACTGAGA	CACTTTAG
CAGAAATG	CAGACCAC	CAGACGCT	CAGAGCCA	CAGAGTGT	CAGCACGA	CAGCATAT	CAGCCTCG
CAGCGAAC	CAGCTAGT	CAGGAGTC	CAGGCTGC	CAGGGTAA	CAGTATTA	CAGTCACC	CAGTCCGT
CATAGCAT	CATAGTTC	CATATAAG	CATATGTT	CATCACCT	CATCCTTT	CATCTCAC	CATCTGCG
CATCTTGA	CATGAGGG	CATGATAC	CATGCGAT	CATGTACC	CATGTCTG	CATTCATA	CATTCGGC
CATTGGAG	CATTGTCT	CCAAACTA	CCAAAGGG	CCAACAAC	CCAACGTT	CCACAAAG	CCACCCGA
CCACGTGG	CCACTGAC	CCAGAAGA	CCAGATCG	CCAGGGTG	CCAGTTTC	CCATAATC	CCATCAGT
CCATCGCC	CCATCTAG	CCATGCAT	CCATGGGA	CCATTCTG	CCCAATAT	CCCACATG	CCCAGCCT
CCCAGTGC	CCCATTTA	CCCCACCC	CCCCATGA	CCCCCGAA	CCCCGGTC	CCCCTAGG	CCCGACGT
CCCGCGGC	CCCGGCTA	CCCGTATT	CCCGTCCG	CCCTCTTC	CCCTTGCT	CCGAGTAG	CCGATAAT
CCGATGCG	CCGCAGGC	CCGCCCTC	CCGCGAGA	CCGCGGAT	CCGCTTCA	CCGGAAAC	CCGGCGAG
CCGGCTTA	CCGGTCGA	CCGTACCA	CCGTCAAA	CCGTCTCT	CCGTGATG	CCGTGCGC	CCGTTGTA

Table 5: Continued.

A set with 1024 code words							
CCTAAAGC	CCTACTCA	CCTAGACG	CCTAGGAC	CCTATGGA	CCTCAATA	CCTCCCCG	CCTCCTAC
CCTCGGCA	CCTCTCGT	CCTGAGCT	CCTGCTGG	CCTGGATC	CCTTCCTT	CGAAAAGT	CGAAACCG
CGAACGCA	CGAAGTAC	CGACATCT	CGACCGGG	CGACTAGA	CGACTGTT	CGACTTAG	CGAGAGAT
CGAGATGC	CGAGCATG	CGAGCTAA	CGATAGTG	CGATGAAG	CGATGCTA	CGATTCGT	CGCACGAG
CGCAGAAA	CGCAGCTG	CGCATACC	CGCCAAGC	CGCCACTT	CGCCCGCT	CGCCGTCA	CGCCTTGT
CGCGGACT	CGCGGTTC	CGCGTCGC	CGCTCGTA	CGCTCTGG	CGCTGTAT	CGCTTCCA	CGCTTGAC
CGGACAGA	CGGACGTC	CGGATCAA	CGGCACAC	CGGCGGTG	CGGCTCCG	CGGGAACA	CGGGCCTT
CGGGGTGG	CGGGTATC	CGGGTTAT	CGGTAGGA	CGGTCCAG	CGGTGGCC	CGGTTACT	CGTAAGCC
CGTACCGT	CGTATTCG	CGTCAGAA	CGTCCATC	CGTCGCGA	CGTCTGGC	CGTGATTA	CGTGCGCG
CGTGGCAC	CGTGTAGT	CGTTAAAC	CGTTCTCC	CGTTGGTT	CTAACTTC	CTAAGACT	CTAATGGC
CTACACGT	CTACATAA	CTACCGTA	CTACGAGC	CTACGGAG	CTACTCCA	CTAGCCAC	CTAGGATA
CTATAGCT	CTATGGTC	CTATTAAA	CTATTTTT	CTCAACGG	CTCAATCC	CTCACTAA	CTCATCAT
CTCCCCTG	CTCCGGGT	CTCGACTC	CTCGCCCA	CTCGGTAG	CTCGTAAC	CTCTCATT	CTCTGCAC
CTCTTGTG	CTGAAGAA	CTGAGGTT	CTGATCTG	CTGCAGCG	CTGCGCCC	CTGCGTTA	CTGCTAAG
CTGGATGT	CTGGCAAT	CTGGCCGG	CTGGGGGA	CTGGTCCT	CTGTACTT	CTGTATAC	CTGTTAGC
CTGTTTCG	CTTAATTT	CTTACGTG	CTTAGCGC	CTTATATC	CTTCACAG	CTTCGTAT	CTTCTACT
CTTCTTTG	CTTGAATG	CTTGACGA	CTTGCAGC	CTTGGCTT	CTTGGTCC	CTTGTGTA	CTTTATCA
CTTTCGAA	CTTTGAGT	CTTTGCCG	CTTTTGCC	GAAAACGT	GAAACCTG	GAAACGGC	GAAAGTGA
GAAATATA	GAAATTAT	GAACAGTG	GAACGAAA	GAAGAACG	GAAGCTAG	GAATACCC	GAATCGAA
GAATGGTT	GAATTACT	GAATTCGG	GACAAGCG	GACATCAA	GACATTTG	GACCAAAT	GACCCCGT
GACCGGTA	GACGACGA	GACGATCC	GACGCGAC	GACGGACA	GACGGCAG	GACGTCCT	GACTAATA
GACTCCTC	GACTCTAT	GACTTGGC	GAGAACAG	GAGACCGA	GAGAGAGG	GAGAGTCC	GAGCCTTA
GAGCGGAG	GAGCTCCA	GAGCTTGC	GAGGCCCC	GAGGGGCT	GAGGTATT	GAGGTGGA	GAGTGCTA
GAGTTAAC	GAGTTGCG	GATAAACT	GATAGGCA	GATATCTC	GATCATTC	GATCCACA	GATCGCCG
GATCGGGC	GATCTATG	GATCTGAT	GATGAGTA	GATGGCGT	GATGTAAA	GATTAAAG	GATTCCCT
GATTCTTG	GATTGATC	GCAAACAC	GCAAGATG	GCAATCCT	GCAATGAA	GCACATAT	GCACCAGC
GCACCGCG	GCACGCAG	GCACGGGT	GCACTATT	GCAGCAAA	GCAGGCGC	GCAGGTCT	GCATAGCA
GCATATGG	GCATTGTC	GCCAAAAA	GCCACTCC	GCCAGGAT	GCCATCGC	GCCCAGAC	GCCCGTCG
GCCCTCAT	GCCGCTGT	GCCGGAGG	GCCGGTAC	GCCGTGAG	GCCTACTT	GCCTCACG	GCCTCGGA
GCCTGGCC	GCCTTTCA	GCGAAACG	GCGAGAAC	GCGAGTTA	GCGATAGA	GCGCAATC	GCGCACGT
GCGCCAAG	GCGGGCAT	GCGGGGTC	GCGGTTGG	GCGTATCC	GCGTCCTG	GCGTGACT	GCGTGGGG
GCGTTGAT	GCTACAGT	GCTACGTC	GCTACTAG	GCTAGCGG	GCTCACTG	GCTCCTGA	GCTCGTTT
GCTGAATT	GCTGACAA	GCTGAGGC	GCTGCGCA	GCTTCCGC	GCTTTCCG	GCTTTTAC	GGAACAAG
GGAACTGT	GGAAGGGG	GGAATAGC	GGACACGA	GGACCATA	GGACCGAT	GGACTCTG	GGAGATTT
GGAGCCCG	GGAGGAGT	GGAGGGTA	GGAGTTGA	GGATAAAA	GGATCTTC	GGATGCAC	GGATTGAG
GGCAATCT	GGCACCGG	GGCAGTAG	GGCATGGA	GGCCCCAA	GGCCCTTT	GGCCTAAC	GGCGAATC
GGCGATGG	GGCGCAAT	GGCGGGCG	GGCGTGTT	GGCTATAC	GGCTGATT	GGCTTAGG	GGGAAATA
GGGAAGAT	GGGACCCT	GGGACTAC	GGGCAGCA	GGGCATAG	GGGCGCGG	GGGCGTTC	GGGCTGGT
GGGGCGTG	GGGGCTCA	GGGGTCAC	GGGTAAGT	GGGTGGAA	GGGTTTTA	GGTAATTG	GGTAGACC
GGTAGCTA	GGTATGAC	GGTCAACG	GGTCACAT	GGTCGGCT	GGTCTCCC	GGTCTGTA	GGTCTTGG
GGTGCCTC	GGTGCGGT	GGTGGATG	GGTGGTGC	GGTGTTCT	GGTTACCA	GGTTAGGG	GGTTCTAA
GGTTTAAT	GTAAAATC	GTAATGTG	GTACCCCC	GTACGGCA	GTACTTTA	GTAGACAT	GTAGAGCC
GTAGCGGA	GTAGGTAA	GTAGTACA	GTAGTCTC	GTATCCGT	GTATCTCA	GTATGACG	GTATGTGC
GTCACAAC	GTCACGTT	GTCAGCTC	GTCAGTCA	GTCCAAGA	GTCCCGAG	GTCCGATG	GTCCTCCG
GTCGAGAA	GTCGCATA	GTCGGTTT	GTCGTAGT	GTCTAACT	GTCTATTG	GTCTGAAA	GTCTGCGG
GTGAAGGC	GTGAGGCG	GTGATACC	GTGATTAA	GTGCACAA	GTGCCAGT	GTGCCGTC	GTGCGCTT
GTGCTTCT	GTGGAAAG	GTGGATTC	GTGGGAGC	GTGTAGTA	GTGTCCAC	GTGTCTTT	GTGTGTAG

Table 5: Continued.

A set with 1024 code words							
GTGTTATG	GTGTTCGA	GTTAACCC	GTTAAGAG	GTTACCAT	GTTACTGC	GTTAGATT	GTTATGCT
GTTCAGTT	GTTCCCTA	GTTCCTCG	GTTCGCAC	GTTCTAGC	GTTGCACT	GTTGGCCA	GTTGGGAT
GTTGTCGG	GTTTATGT	GTTTCAGA	GTTTGGTG	GTTTTCTT	TAAAATTG	TAAACAAT	TAAAGACG
TAAAGGTC	TAACCGAG	TAACGTGC	TAACTACC	TAACTGGT	TAAGACCT	TAAGAGAC	TAAGCTTC
TAAGGAGA	TAAGTCAA	TAAGTGTG	TAATACTA	TAATCCAC	TAATCTCG	TAATTAAG	TACAAACA
TACAACTC	TACAGGAG	TACATAGT	TACCATGT	TACCCAGA	TACCTGAA	TACGAAGC	TACGCAAG
TACGGTCT	TACGTCGG	TACGTGCC	TACTCGCT	TACTGATG	TACTGCCA	TACTTCTT	TAGACATC
TAGATCCG	TAGCACTG	TAGCGATT	TAGCTTAG	TAGGAAAT	TAGGAGCG	TAGGTCTC	TAGGTTGT
TAGTAAGA	TAGTCGGG	TAGTGGAT	TAGTGTTC	TAGTTTCA	TATAAGAT	TATAATGC	TATACGTA
TATACTCT	TATATTAA	TATCAAAA	TATCCCCC	TATCCTGG	TATCGGTG	TATCGTCA	TATGACAG
TATGCATT	TATGCCGA	TATTATTT	TATTGCGG	TATTTGTC	TCAAATGA	TCAACCGC	TCAAGAAA
TCAAGGCT	TCAATTCG	TCACAAGT	TCACACAA	TCACGATC	TCACTCGG	TCAGAACC	TCAGAGTT
TCAGCCTG	TCAGTAAT	TCAGTGCA	TCATCCCT	TCATCTTA	TCATGGAG	TCATTTGT	TCCAAGGC
TCCACACT	TCCATCTG	TCCCCCTT	TCCCCTAG	TCCCGCGC	TCCGATTC	TCCGGGAA	TCCGTGGT
TCCTACAG	TCCTCGAC	TCCTGTTT	TCCTTAGA	TCGAAATT	TCGACCCA	TCGAGCTC	TCGATCGT
TCGATTAC	TCGCCATA	TCGCCTGT	TCGCGGCC	TCGGACGG	TCGGATCA	TCGGCACG	TCGGCCAC
TCGGCGGA	TCGGGTGC	TCGTAGTG	TCGTATAT	TCGTGCCG	TCGTTATC	TCTACGCG	TCTAGTAT
TCTATAGG	TCTCAGAG	TCTCATCC	TCTCCGGC	TCTCTAAC	TCTCTGCT	TCTGCTAA	TCTGGACT
TCTGTCGC	TCTGTTTG	TCTTAACA	TCTTACGT	TCTTCAAT	TCTTGCAC	TCTTGGTA	TGAAAGTA
TGAAGTTT	TGAATCGA	TGAATGAT	TGACCACG	TGACGCCC	TGACGGAA	TGACTTCA	TGAGCAAC
TGAGGGGC	TGAGGTCG	TGATACGG	TGATAGCC	TGATCGTT	TGATGACT	TGATTTAC	TGCAAATG
TGCACTAT	TGCAGCAC	TGCAGTGA	TGCCAGCG	TGCCCTGC	TGCCGAAT	TGCCTCTC	TGCGACAA
TGCGCGGG	TGCGCTTA	TGCGGCTT	TGCGTACG	TGCTCACC	TGCTGGTC	TGCTTTCT	TGGACGAA
TGGACTGG	TGGATAAG	TGGCATTA	TGGCCCAT	TGGCGAGC	TGGCGTCT	TGGCTGAC	TGGGAGGT
TGGGGCGA	TGGGGGAG	TGGTACCT	TGGTCCGC	TGGTGATA	TGGTTCTG	TGTAACTT	TGTACACA
TGTAGAGT	TGTAGCCG	TGTCACGC	TGTCCCTG	TGTCGTAC	TGTCTATT	TGTGAAGA	TGTGCCCT
TGTGGGCA	TGTGTCTA	TGTTATAG	TGTTCAGG	TGTTTGCG	TGTTTTGA	TTAACCAA	TTAAGCGG
TTAAGTCC	TTACAACA	TTACAGTC	TTACCATT	TTACCTGA	TTACGTTG	TTACTTAT	TTAGACGC
TTAGATTA	TTAGCGCT	TTAGTAGG	TTATAAAT	TTATCAGC	TTATTCCG	TTATTGTA	TTCACTTG
TTCAGAGC	TTCATATA	TTCATGCG	TTCCAAAC	TTCCCGCA	TTCCGCAA	TTCCTTGG	TTCGAATT
TTCGCCAT	TTCGGCCC	TTCGGGTG	TTCTAGGG	TTCTATAA	TTCTCCTA	TTCTCTGT	TTCTGTCG
TTCTTGAT	TTCTTTTC	TTGAACGA	TTGACGGT	TTGAGGAC	TTGATTTT	TTGCAGAT	TTGCATGC
TTGCCCCG	TTGCGGGG	TTGCTCTA	TTGGCTCC	TTGGGTAT	TTGGTCAG	TTGGTGGC	TTGTAACC
TTGTCAAG	TTGTGTGA	TTTAATCG	TTTAGTTA	TTTATAAT	TTTCAAGG	TTTCCTTC	TTTCGCGT
TTTCTGGA	TTTGCGAG	TTTGGAAA	TTTGGTGG	TTTGTTAC	TTTTACTC	TTTTGGCT	TTTTTCAA

become a burden for the communication media and which also poses also a challenge for hiding strategies of large messages. The steganographic approach including message distribution and the selection of inconspicuous dissemination venues must be carefully analyzed. For example, large encrypted messages encoded as long *in silico* DNA sequences can be better hidden in databases for DNA coding sequences, DNA contigs or mRNA sequences, while relatively short messages would be better hidden as DNA and RNA primer sequences or as microarray probes.

One potential weakness of the current approach could stem from peculiarities of the language in which the original message was written, assuming that the attacker has already guessed it. For example, if English is the language, then an analysis based on occurrences of double letters such as double Ls in a fairly limited number of words could be used to find partial (code word, character) associations. A potential extension inspired from the Belasso Ciphers [37], which were later wrongfully attributed to Vigenère [38], that will add confusion and increased security to HyDEn is to encode each character with multiple code words selected uniformly at random, without breaking the error detection and correction capabilities of the DNA code. Table 5 presents an $A_4(8,3)$ code with 1024 DNA sequences of length 8 and minimum pairwise Hamming distance 3, which could be used as a replacement of the code from Table 2. Each extended ASCII character could be encoded using one out of 4 different code words, each selected with equal probability. Lower (2048) and upper (2340) bounds published by Bogdanova et al. [39] and hosted on Dr. Andries Brower's website [40] suggest that even larger $A_4(8,3)$ DNA codes can be generated.

6. Conclusion

Here, we have presented a novel stegano-cryptographic approach called HyDEn (hybrid DNA encryption), which uses custom-built error-correcting DNA Hamming codes, a randomized code assignment procedure and cyclic permutations based on a private key. HyDEn represents a symmetric cipher that is capable of encrypting and disguising information as long DNA sequences in public bioinformatics discussion groups and DNA sequence databases. Our cryptosystem has significant error tolerance and adds another dimension to the information security field. We are currently working on experimentally evaluating and further improving HyDEn's capabilities following the ideas described in Section 5.3.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments that led to the improvement of this paper. They are grateful to their colleagues from the Knowledge Discovery and the Learning and Collaborative Technologies Groups for helping in reviewing and improving this paper. Funding for this research was provided by the National Research Council Canada.

References

[1] F. Miller, *Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams*, C.M. Cornwell, 1882.

[2] D. Coppersmith, "Data Encryption Standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243–250, 1994.

[3] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[5] P. Zimmermann, *PGP Source Code and Internals*, MIT Press, Cambridge, Mass, USA, 1995.

[6] C. H. Huang, S. C. Chuang, and J. L. Wu, "Digital invisible ink and its applications in steganography," in *Proceedings of the 8th Workshop on Multimedia and Security (MM&Sec '06)*, pp. 23–28, ACM, New York, NY, USA, September 2006.

[7] E. Cole, *Hiding in Plain Sight: Steganography and the Art of Covert Communication*, John Wiley & Sons, New York, NY, USA, 1st edition, 2003.

[8] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann, San Francisco, Calif, USA, 2nd edition, 2007.

[9] C. T. Clelland, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots," *Nature*, vol. 399, no. 6736, pp. 533–534, 1999.

[10] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An overview of image steganography," in *ISSA 2005 New Knowledge Today Conference*, J. H. P. Eloff, L. Labuschagne, M. M. Eloff, and H. S. Venter, Eds., pp. 1–11, ISSA, Pretoria, South Africa, 2005.

[11] B. Anckaert, B. D. Sutter, D. Chanet, and K. D. Bosschere, "Steganography for executables and code transformation signatures," in *Proceedings of the 7th International Conference on Information Security and Cryptology (ICISC '04)*, pp. 425–439, December 2004.

[12] B. Anam, K. Sakib, M. A. Hossain, and K. P. Dahal, *Review on the Advancements of DNA Cryptography*, CoRR, 2010.

[13] V. I. Risca, "DNA-based steganography," *Cryptologia*, vol. 25, pp. 37–49, 2001.

[14] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 26, pp. 147–160, 1950.

[15] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing, Amsterdam, The Netherlands, 2nd edition, 1978.

[16] N. Aboluion, D. H. Smith, and S. Perkins, "Linear and nonlinear constructions of DNA codes with Hamming distance d, constant GC-content and a reverse-complement constraint," *Discrete Mathematics*, vol. 312, no. 5, pp. 1062–1075, 2012.

[17] R. Montemanni and D. H. Smith, "Construction of constant GC-content DNA codes via a variable neighbourhood search algorithm," *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 3, pp. 311–326, 2008.

[18] D. C. Tulpan, H. H. Hoos, and A. E. Condon, "Stochastic local search algorithms for DNA word design," in *DNA Computing*, vol. 2568 of *Lecture Notes in Computer Science*, pp. 229–241, 2003.

[19] P. Gaborit and O. D. King, "Linear constructions for DNA codes," *Theoretical Computer Science*, vol. 334, no. 1–3, pp. 99–113, 2005.

[20] O. D. King, "Bounds for DNA codes with constant GC-content," *Electronic Journal of Combinatorics*, vol. 10, article 13, 2003.

[21] A. Marathe, A. E. Condon, and R. M. Corn, "On combinatorial DNA word design," *Journal of Computational Biology*, vol. 8, no. 3, pp. 201–219, 2001.

[22] A. Leier, C. Richter, W. Banzhaf, and H. Rauhe, "Cryptography with DNA binary strands," *BioSystems*, vol. 57, no. 1, pp. 13–22, 2000.

[23] M. Hirabayashi, H. Kojima, and K. Oiwa, "Design of true random one-time pads in DNA XOR cryptosystem," *Natural Computing*, vol. 2, pp. 174–183, 2010.

[24] A. Gehani, T. Labean, and J. Reif, "DNA-based cryptography," in *Proceedings of the 5th DIMACS Workshop on DNA Based Computers*, MIT, American Mathematical Society, 1999.

[25] M. Arita and Y. Ohashi, "Secret signatures inside genomic DNA," *Biotechnology Progress*, vol. 20, no. 5, pp. 1605–1607, 2004.

[26] D. C. Tulpan and H. H. Hoos, "Hybrid randomised neighbourhoods improve stochastic local search for DNA code design," in *Advances in Artificial Intelligence*, vol. 2671 of *Lecture Notes in Computer Science*, pp. 418–433, 2003.

[27] P. C. Wong, K. Wong, and H. Foote, "Organic data memory using the DNA approach," *Communications of the ACM*, vol. 46, no. 1, pp. 95–98, 2003.

[28] R. K. Saiki, D. H. Gelfand, S. Stoffel et al., "Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase," *Science*, vol. 239, no. 4839, pp. 487–491, 1988.

[29] D. Heider and A. Barnekow, "DNA-based watermarks using the DNA-Crypt algorithm," *BMC Bioinformatics*, vol. 8, article 176, 2007.

[30] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, New York, NY, USA, 4th edition, 2002.

[31] B. Schneier, "Description of a new variable-length key, 64-bit block cipher(blowfish)," in *Fast Software Encryption, Cambridge Security Workshop*, pp. 191–204, Springer, London, UK, 1994.

[32] H. J. Shiu, K. L. Ng, J. F. Fang, R. C. T. Lee, and C. H. Huang, "Data hiding methods based upon DNA sequences," *Information Sciences*, vol. 180, no. 11, pp. 2196–2208, 2010.

[33] M. R. N. Torkaman, N. S. Kazazi, and A. Rouddini, "Innovative approach to improve hybrid cryptography by using DNA steganography," *International Journal on New Computer Architectures and Their Applications*, vol. 202, pp. 225–236, 2012.

[34] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, pp. 675–701, 1937.

[35] F. W. Kasinsiki, *Die Geheimschriften und die Dechiffrir-Kunst*, E.S. Mittler und Sohn, Berlin, Germany, 1863.

[36] J. Stirling, Methodus differentialis, sive tractatus de summation et interpolation serierum infinitarium, 1730.

[37] G. B. Belasso, La cifra del sig. giovan battista bellaso, gentil huomo bresciano, nuovamente da lui medesimo ridotta à grandissima brevità et perfettione, 1553.

[38] B. D. Vigenère, *Traicté des chiffres, ou Secrètes manières d'escrire*, Abel L'Angelier, Paris, France, 1st edition, 1587.

[39] G. T. Bogdanova, A. E. Brouwer, S. N. Kapralov, and P. R. J. Östergård, "Error-correcting codes over an alphabet of four elements," *Designs, Codes, and Cryptography*, vol. 23, no. 3, pp. 333–342, 2001.

[40] A. Brouwer, "Table of general quaternary codes," 2001, http://www.win.tue.nl/~aeb/codes/quaternary-1.html.