

# 基于梅尔频率倒谱系数与动态时间规整的安卓声纹解锁系统

陈锦飞, 徐 欣

(杭州电子科技大学 通信工程学院, 杭州 310018)

**摘 要:** 安卓设备通常采用数字或图形密码解锁, 但此类口令形式的密码安全性不高, 而且部分安卓版本存在锁屏绕过漏洞的问题。为此, 设计一种利用用户声纹特征的安卓解锁系统。采用梅尔频率倒谱系数提取声纹特征, 使用动态时间规整算法进行文本相关的声纹模式匹配, 并结合安卓 NDK 技术实现快速声纹识别。实验结果表明, 该系统具有较高的解锁成功率和较快的解锁速度, 相比数字或图形解锁, 声纹解锁安全性更高, 用户体验更好。

**关键词:** 梅尔频率倒谱系数; 动态时间规整; 声纹识别; 安卓系统; 声纹解锁

**中文引用格式:** 陈锦飞, 徐 欣. 基于梅尔频率倒谱系数与动态时间规整的安卓声纹解锁系统[J]. 计算机工程, 2017, 43(2): 201-205.

**英文引用格式:** Chen Jinfei, Xu Xin. Android Voiceprint Unlocking System Based on Mel Frequency Cepstrum Coefficient and Dynamic Time Warping[J]. Computer Engineering, 2017, 43(2): 201-205.

## Android Voiceprint Unlocking System Based on Mel Frequency Cepstrum Coefficient and Dynamic Time Warping

CHEN Jinfei, XU Xin

(College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China)

**[Abstract]** Android devices usually use digital or graphical password for unlocking screen, but such form of password is not safe enough, and some versions of Android exist keyguard bypass vulnerability. Therefore, this paper designs an Android keyguard system which uses user's unique voiceprint feature to unlock screen. This paper uses Mel Frequency Cepstrum Coefficient (MFCC) as the voiceprint feature and Dynamic Time Warping (DTW) for voiceprint pattern matching, and uses Android NDK technology to achieve quick voiceprint recognition. Experimental results show that this voiceprint keyguard system has a good unlocking success rate and unlocking speed, and compared with digital or graphical unlocking, unlocking screen with voiceprint is safer and gives users a better experience.

**[Key words]** Mel Frequency Cepstrum Coefficient (MFCC); Dynamic Time Warping (DTW); voiceprint recognition; Android system; voiceprint unlocking

**DOI:** 10.3969/j.issn.1000-3428.2017.02.033

## 0 概述

Android 已经成为当今流行的移动操作系统。根据 IDC 智能手机出货量数据显示, Android 设备出货量达到 10.59 亿部, 市场份额高达 81.5%<sup>[1]</sup>, 但是随着安卓设备的广泛使用, 安卓系统的安全问题也日益凸显。锁屏作为设备安全的第一道大门, 通常使用数字或图形密码解锁, 容易被窥探或暴力破解, 安全性较低<sup>[2]</sup>。除此之外, 为提高设备安全性, 生物识别技术也逐渐应用于移动设备的锁屏认证, 如指纹识别、人脸识别。指纹识别成功率高、识别速度快, 但需要专门的指纹硬件提取模块, 成本较高<sup>[3]</sup>。

人脸识别借助摄像头提取人脸特征, 成本较低, 但受外界因素影响较大, 导致识别成功率不高, 且识别算法十分复杂<sup>[4]</sup>。

声纹如同指纹、人脸一样, 是人体特有的一种生物特征。声纹识别技术是生物识别技术的一种, 基于个体独一无二的声纹特征, 相比传统密码更加安全, 同时声纹识别技术借助麦克风提取声纹特征, 由于其准确性、便捷性以及经济性, 开始在众多身份认证领域得到广泛使用<sup>[5]</sup>。因此, 本文利用声纹特征提取技术和模式匹配算法, 使得安卓声纹解锁在保证设备安全性的同时, 能兼顾解锁成功率和解锁速度。

**作者简介:** 陈锦飞 (1990—), 男, 硕士研究生, 主研方向为信息安全; 徐 欣, 教授。

**收稿日期:** 2016-03-08 **修回日期:** 2016-04-29 **E-mail:** jeffy26@163.com

## 1 声纹解锁系统

安卓声纹解锁系统由 3 个模块组成,如图 1 所示,分别是声纹提取模块、声纹匹配模块和锁屏模块。与非线性系统类似,信号频率不同,人耳的响应灵敏度也不同,梅尔频率倒谱系数(Mel Frequency Cepstrum Coefficient, MFCC)考虑到人类的听觉特征,先将线性频谱映射到基于听觉感知的 Mel 非线性频谱中,然后转换到倒谱上<sup>[6]</sup>。线性预测倒谱系数(Linear Prediction Cepstrum Coefficient, LPCC)也常用来提取声纹特征,它基于声道模型,但鲁棒性比 MFCC 差。MFCC 更符合人耳听觉特性,在信噪比降低时,识别性能也能处于较好的水平<sup>[7]</sup>。

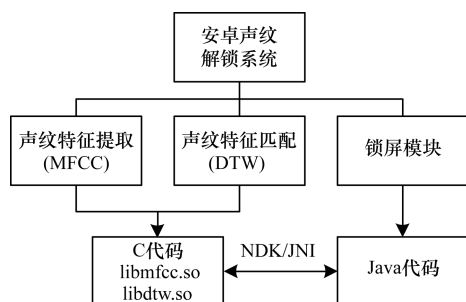


图 1 声纹解锁系统框架

在语音识别中,基于动态规划思想的动态时间规整(Dynamic Time Warping, DTW)算法是出现较早且简单有效的经典语音识别算法<sup>[8-9]</sup>。为提升安卓声纹解锁系统的解锁速度,从而提高用户体验,选择 DTW 对用户声纹特征 MFCC 进行模式匹配。隐马尔可夫模型(Hidden Markov Model, HMM)和高斯混合模型(Gaussian Mixture Model, GMM)可能有更好的抗噪性能,但为了能得到精确的模型参数,需要大量的语音训练数据并进行复杂的计算。而对 DTW 算法而言,训练中只需少量的计算<sup>[10]</sup>。在文本相关的声纹识别中,DTW 算法具有更快的识别速度,识别准确度也较高。

NDK(Native Development Kit)是 Google 为安卓开发者提供的工具包,使得开发者能够利用这些工具快速开发 C/C++ 的动态库,在生成应用 APK 文件时,动态库和 Java 程序也会自动打包<sup>[11]</sup>。为提高声纹特征提取以及声纹模式匹配的速度,均使用 C 代码实现 MFCC 提取和 DTW 模式匹配,并通过 NDK 打包成动态链接库 libmfcc.so 和 libdtw.so。系统框架及锁屏模块由 Java 代码实现,通过 JNI(Java Native Interface)调用 C 动态库实现快速声纹特征提取和模式匹配。

## 2 MFCC 参数提取过程

MFCC 是一种能够充分利用人耳感知特性的参数。MFCC 和线性频率的转换关系如下:

$$f_{\text{mel}} = 2595 \lg \left[ 1 + \frac{f}{700} \right]$$

MFCC 提取过程如图 2 所示。

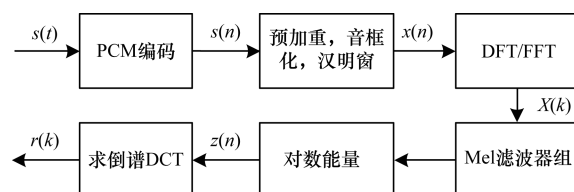


图 2 MFCC 提取过程

在图 2 中,预加重是对语音信号做差分处理,以消除非语音信号和静默语音信号。 $S_1(n)$  ( $n = 0, 1, \dots, N-1$ ) 表示时域信号,预增强公式为:

$$S(n) = S_1(n) - a \times S_1(n-1), 0.9 < a < 1.0$$

在实验过程中,  $a$  取 0.95。音框化即对语音信号分帧,便于后续处理。在输入语音流的分帧处理上,音框间的重迭单位采用 128,而音框单位(帧)采用 256 个采样点。汉明窗即对每帧信号加窗,以减小吉布斯效应的影响。假设音框化的信号( $M$  帧共  $N$  点)为  $S(n)$ ,  $n = 0, 1, \dots, N-1$ , 那么乘上汉明窗后为:

$$x(n) = S(n) \times W(n)$$

其中,窗函数选取如下:

$$W(n) = W(n, a) = (1 - a) - a \cos \left( \frac{2\pi n}{N-1} \right)$$

$$0 \leq n \leq N-1; a = 0.46$$

为得到信号的频谱系数  $X(k)$ ,对输入语音信号预处理后,需要进行离散傅里叶变换(Discrete Fourier Transform, DFT),计算公式为:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, 0 \leq k \leq N$$

为模拟人耳的掩蔽效应,Mel 滤波器组覆盖范围越近似于人耳的临界带宽,效果越好<sup>[12]</sup>。 $\text{melf}_8[]$ ,  $\text{melf}_{16}[]$  分别为 8 kHz 和 16 kHz 条件下的滤波器数组。

$\text{melf}[]$  滤波器数组以结构化方式存储,用稀疏矩阵  $\text{melf}[20][129]$  表示。 $z[20]$  为每个滤波器输出能量的对数表示,计算公式为:

$$z[] = \ln(\text{melf}[] \times (|X(0:128)|^2))$$

对对数能量进行离散余弦变换(Discrete Cosine Transform, DCT),得到 DCT 系数数组  $r[20]$ ,即:

$$r[] = \text{dct}(z[])$$

$$r[k] = \sum_{l=0}^{D-1} z[l] \cos \left( \frac{\pi(2l+1)k}{2D} \right), D = 20$$

通过上式计算每一个  $k$  值对应的  $r[k]$ ,即可求得各帧语音信号的 MFCC 参数,所有语音信号帧的参数均通过此公式计算并保存。

## 3 DTW 算法原理

DTW 算法原理如图 3 所示。在图 3 所示网格中,纵轴  $m(1 \sim M)$  表示参考模板的各帧,横轴  $n(1 \sim N)$  表示测试模板的各帧。网格交叉点  $(T(n))$ ,

$R(m)$ ) 代表参考模版中某一帧与测试模版中某一帧的交汇。

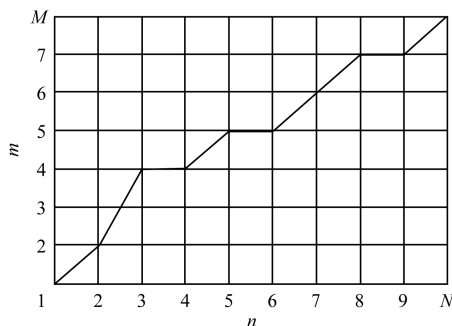


图 3 DTW 算法原理示意图

DTW 算法需要进行 2 步操作: 1) 计算参考模板和测试模板各帧之间的欧氏距离, 得到帧匹配距离矩阵; 2) 在该矩阵中找出一条最佳路径<sup>[13]</sup>。最佳路径搜索过程具体如下:

从点  $(1, 1)$  开始搜索, 对于局部路径约束, 点  $(T(n-1), Rm)$ ,  $(T(n-1), R(m-1))$  和  $(T(n), R(m-1))$  是能够到达点  $(T(n), R(m))$  的唯一可选前续格点。为得到最佳路径,  $(T(n), R(m))$  的前续格点必须是距离自己最近的那个点, 与之对应的路径累积距离为:  $D(T(n), R(m)) = d(T(n), R(m)) + \min \{ D(T(n-1), R(m)), D(T(n-1), R(m-1)), D(T(n), R(m-1)) \}$ 。从点  $(1, 1)$  开始 (令  $D(1, 1) = 0$ ) 搜索, 反复递推到点  $(N, M)$ , 即可找出最佳匹配路径 (最优路径), 该路径所对应的匹配距离为  $D(N, M)$ <sup>[14]</sup>。在进行声纹识别时, 将之前训练得到的 MFCC 模板与测试 MFCC 模板进行一一匹配, 若得到的最小匹配距离  $D_{\min}(N, M)$  在给定阈值内, 则声纹匹配成功。

## 4 安卓锁屏实现

安卓锁屏模块由以下步骤实现:

1) 开启后台服务, 注册广播接收器 (Broadcast Receiver) 来监听屏幕开关事件。

屏幕开关时安卓系统会发出以下 2 个广播事件:

- (1) 屏幕开启广播: `android.intent.action.SCREEN_ON`;
- (2) 屏幕熄灭广播: `android.intent.action.SCREEN_OFF`。

接收对应的广播来调用锁屏界面。为在屏幕开启时立刻加载自定义声纹锁屏界面, 后台服务应该在监听到屏幕熄灭广播时调用锁屏界面, 这样当屏幕唤醒时就可立刻开始声纹验证, 无需等待锁屏视图加载。

2) 屏蔽系统锁屏

替换系统锁屏要调用 `KeyguardManager` 类相应方法解除屏幕锁定, 在应用 `AndroidManifest.xml` 文件时添加以下权限申明:

```
<uses-permission
android:name="android.permission.DISABLE_KEYGUARD"/>
```

屏蔽系统锁屏代码为:

```
screenLockManag = (KeyguardManager) context.getSystemService(
Context.KEYGUARD_SERVICE);
mScreenLock = screenLockManag.newKeyguardLock("");
mScreenLock.disableKeyguard();
```

3) 屏蔽系统 Home 键、Back 键、Menu 键。

锁屏界面只有当声纹匹配成功后才能退出, 不能对除电源键以外的物理按键进行响应, 即必须屏蔽 Android 标准的 3 个功能键: Home, Back, Menu。

(1) 屏蔽 Menu 键重写 `onCreateOptionsMenu()` (Menu menu) 方法, 不加载 menu.xml 文件即可。

(2) 屏蔽 Back 键重写 `onBackPressed()` 方法, 不调用 `super.onBackPressed()` 即可。

(3) 屏蔽 Home 键, Android 4.0 以下可以在 activity 中添加下列代码:

```
getWindow().addFlags(WindowManager.LayoutParams.
FLAG_HOMEKEY_DISPATCHED)
```

由于 Android 4.0 及以上的系统在 Framework 层就将 Home 键信号截住, 因此无法在应用层屏蔽 HOME 键, 但可以采用加载锁屏视图 view 的方式, 提高 view 优先级, 让锁屏 view 覆盖其他系统 view, 从而达到屏蔽 Home 键的锁屏效果, 具体代码如下:

```
WindowManager.addView(lockView, param)
```

其中, lockView 为锁屏界面视图; param 为该视图的配置参数。关键参数 type 必须配置为:

```
param.type = LayoutParams.TYPE_SYSTEM_ALERT
param.type = LayoutParams.TYPE_SYSTEM_ERROR
```

同时, 必须在应用配置文件中添加如下权限:

```
<uses-permission android:name="android.permission.
SYSTEM_ALERT_WINDOW"/>
```

## 5 NDK 开发过程

安卓 API 类库提供了丰富的控件, 如 Button (按钮)、TextView (文本框) 等用于实现用户交互, 系统 UI 只需在 layout 中编写相应的 XML 文件即可, 对性能要求不高, 因此, 整个声纹解锁系统的框架和锁屏模块由 Java 代码实现。声纹特征提取和模板匹配均需要较大的运算量, 因此, 最好由 C/C++ 编译成本地代码实现, 以提高解锁速度和用户体验。

JNI 提供若干 API 实现 Java 和其他语言的通信 (主要是 C/C++)。Java 代码可以借助 JNI 调用非 Java 语言编写的代码<sup>[15]</sup>。Google 发布 Android SDK 时, JNI 编程方式就能在 Dalvik 虚拟机上实现, 第三方应用可以编写自己的 C/C++ 动态库, 然后通过 JNI 方式调用。“Java + C/C++” 的编程方式可在 Android 平台上实现。在 NDK 发布后, “Java + C” 成为官方支持的开发方式。NDK 中包含交叉编译器, 为了隔离 ABI、平台、CPU 等差异, 开发者只需对 mk 文件作简单修改, 即在文件中指明编译特性及哪些文件

需要编译等即可得到 so 动态库文件<sup>[16]</sup>。Java 应用和动态库可以通过 NDK 自动打包成应用 APK 文件。

NDK 开发主要涉及以下内容:

1) 在安卓程序 activity 类中声明需要用到的本地接口:

```
public class voiceLock extends Activity
{
    //load native library
    public native void mfccJNI(); public native void dtwJNI();
    public void onCreate( Bundle savedInstanceState )
    {
        //调用 mfccJNI(), dtwJNI()
    }
}
```

其中, mfccJNI() 和 dtwJNI() 来自 libmfcc. so 和 libdtw. so 的本地接口, so 动态库由 System. loadLibrary() 加载:

```
static { System. loadLibrary( "mfcc" );
        System. loadLibrary( "dtw" ); }
```

2) 进入安卓工程 jni 目录, 使用 javah 生成 C 代码头文件:

```
javah-classpath bin-d jni com. example. voice lock. voiceLock
```

3) 编写 mfcc. c 和 dtw. c, 本地接口 mfccJNI() 和 dtwJNI() 应声明为:

```
void Java_com_example_voicelock_voiceLock_mfccJNI
( JNIEnv * env, jobject this )
void Java_com_example_voicelock_voiceLock_dtwJNI
( JNIEnv * env, jobject this )
```

4) 为指定源文件的编译规则, 需要编写 Android 的 Makefile 文件, 即 Android. mk, 文件路径位于 JNI 目录下。主要声明共享库的名称 LOCAL\_MODULE 和声明 C 源文件 LOCAL\_SRC\_FILES, 然后利用 Android NDK 开发包中的 ndk-build 脚本生成对应的 libmfcc. so 和 libdtw. so 共享库。

5) 在 eclipse 中刷新 voicelock 工程, 重新编译生成安卓应用安装包 apk 文件, libmfcc. so 和 libdtw. so 共享库一起打包在 apk 文件中。此时, 声纹解锁系统 apk 开发完成。

## 6 实验与结果分析

整个声纹解锁系统流程如图 4 所示。声纹解锁系统基于 Android4. 4. 2 SDK 开发, 测试设备为运行 Android4. 4. 2 的安卓手机 (2 GB 内存, 1. 6 GHz 四核 CPU)。录音采用手机麦克风, 输入语音流采用单声道、16 bit、11. 025 kHz 采样, 提取声纹 MFCC 参数存储在 SD 卡上, 开启锁屏后进入监听模式, 亮屏后点击麦克风进行声纹匹配, 匹配成功即可退出锁屏界面, 系统运行界面如图 5 所示。

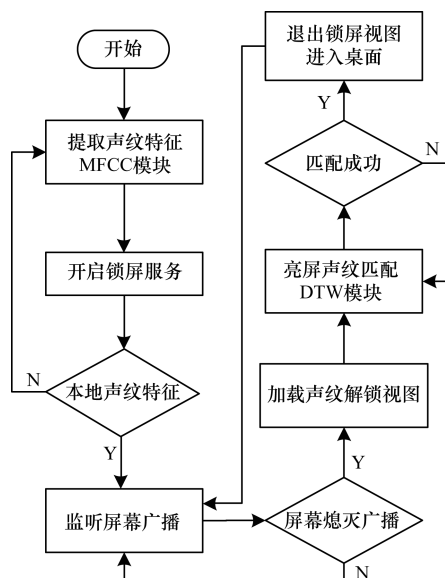


图 4 声纹解锁系统流程



(a)系统主界面

(b)锁屏界面

图 5 系统运行界面

为提高解锁速度, 提高用户体验, 采用文本相关的声纹识别, 因此, 用户可以定义四字短语用于声纹提取和识别。在提取 MFCC 参数时, 需要连续朗读一定次数的四字短语, 且均用普通话朗读。

实验选取了 3 个四字短语, 提取声纹特征时训练 5 次, 并在室内和室外分别进行 100 次解锁测试。测试解锁成功率和平均解锁时间, 解锁时间为说出四字短语到解锁成功返回系统桌面所用时间, 结果如表 1 所示。

表 1 声纹解锁系统实验结果 1

测试文本	解锁成功率/%		平均解锁时间/s	
	室内	室外	室内	室外
芝麻开门	89	82	3. 25	3. 23
声纹解锁	87	79	3. 17	3. 18
新年快乐	88	80	3. 21	3. 19

由以上结果可以看出, 声纹识别成功率受文本影响不大, 受识别环境影响较大, 室内识别成功率约 88%, 室外识别成功率约 80%。由此可知, 由于提取声纹特征均在室内进行, 因此在室外环境噪声较大时, 识别成功率会明显下降。通过提高声纹特征提

取训练次数,得到室外室内各提取 5 次声纹特征的识别成功率如表 2 所示。

表 2 声纹解锁系统实验结果 2

测试文本	解锁成功率/%		平均解锁时间/s	
	室内	室外	室内	室外
芝麻开门	87	85	3.21	3.19
声纹解锁	85	84	3.18	3.21
新年快乐	86	86	3.23	3.22

可以看出,通过提高训练次数,在室内外分别提取声纹特征后,室外声纹识别成功率提高到 85% 左右,室内识别成功率较之前有小幅下降,在 86% 左右。平均解锁时间均为 3.2 s 左右,在理想情况下,用户讲完声纹文本口令一秒内即可解锁成功。解锁时间不受文本和环境的影响,主要与 MFCC 算法、DTW 算法以及硬件平台计算能力有关。

## 7 结束语

本文介绍了 Android 平台下基于 MFCC 和 DTW 算法的声纹识别系统,并将其与安卓系统锁屏相结合,实现利用声纹解锁手机的目的。相比传统数字或图形解锁方式,声纹解锁不易破解,安全性更高。在 Android 4.4.2 平台上的测试结果表明,该系统具有较高的解锁成功率和解锁速度。下一步将重点研究语音降噪算法,提高声纹解锁系统的识别成功率和用户体验。

### 参考文献

- [1] IDC. Android and IOS Squeeze the Competition[EB/OL]. [2016-03-05]. <http://www.idc.com/getdoc.jsp?containerId=prUS25450615>.
- [2] 王即墨,计超豪,裴洪卿,等. Android 智能手机锁屏密码及破解方法研究[J]. 刑事技术,2015,25(2):142-145.
- [3] 张圆圆. 指纹识别技术相关算法的研究[D]. 北京:

北京邮电大学,2012.

- [4] 王二伟. 基于 Android 平台人脸检测与识别研究[D]. 西安:西安电子科技大学,2013.
- [5] Gunson N, Marshall D, McInnes F, et al. Usability Evaluation of Voiceprint Authentication in Automated Telephone Banking: Sentences Versus Digits[J]. Interacting with Computers,2011,23(1):57-69.
- [6] Nakagawa S, Wang L, Ohtsuka S, et al. Speaker Identification and Verification by Combining MFCC and Phase Information[J]. IEEE Transactions on Audio, Speech, and Language Processing,2012,20(4):1085-1095.
- [7] 郭春霞. 基于 MFCC 的说话人识别系统研究[D]. 西安:西安电子科技大学,2006.
- [8] 李正欣,张凤鸣,李克武,等. 基于 DTW 的多元时间序列模式匹配方法[J]. 模式识别与人工智能,2011,24(3):425-430.
- [9] 胡金平,陈若珠,李战明,等. 语音识别中 DTW 改进算法的研究[J]. 微型机与应用,2011,30(3):30-32.
- [10] 宋 艳. 基于嵌入式语音识别系统的研究[D]. 西安:西安科技大学,2011.
- [11] Google Inc.. Android NDK Reference [EB/OL]. [2016-03-05]. <http://developer.android.com/tools/sdk/ndk/index.htm>.
- [12] 王华朋,杨洪臣. 声纹识别特征 MFCC 的提取方法研究[J]. 中国人民公安大学学报(自然科学版),2008,14(1):28-30.
- [13] Jeong Y S, Jeong M K, Omitaomu O A, et al. Weighted Dynamic Time Warping for Time Series Classification[J]. Pattern Recognition,2011,44(9):2231-2240.
- [14] 朱淑琴,赵 瑛. DTW 语音识别算法研究与分析[J]. 微计算机信息,2012,23(5):150-151.
- [15] 张华平,玄光哲,于贵平,等. 基于 JNI 技术应用框架的分析和实现[J]. 吉林大学学报(信息科学版),2003,21(2):188-191.
- [16] 王有禄,李代平. Android 系统下基于 NDK 方式的图形开发[J]. 计算机系统应用,2012,25(12):56-59.

编辑 陆燕菲

(上接第 200 页)

### 参考文献

- [1] 许 佳,周丹平,顾海东. APT 攻击及其检测技术综述[J]. 保密科学技术,2014(1):34-40.
- [2] 王 宇,韩伟杰. APT 攻击特征分析与对策研究[J]. 保密科学技术,2013(12):32-43.
- [3] 云晓春. 国家互联网网络安全宏观态势综述[J]. 保密科学技术,2012(1):6-8.
- [4] 刘 昕. 大数据背景下的 APT 攻击检测与防御[J]. 电子测试,2014(1):80-81.
- [5] Yen T F, Oprea A, Onarlioglu K, et al. Beehive: Large-scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks[C]//Proceedings of the 29th Annual Computer Security Applications Conference. New York, USA:ACM Press,2013:199-208.
- [6] 戴阳阳,李朝锋,徐 华. 初始点优化与参数自适应的密度聚类算法[J]. 计算机工程,2016,42(1):203-209.
- [7] 陈 燕. 对两个经典聚类算法的分析[J]. 现代电子技

术,2007,30(17):174-176.

- [8] 冯少荣,肖文俊. DBSCAN 聚类算法的研究与改进[J]. 中国矿业大学学报,2008,37(1):105-111.
- [9] 刘 萍,郑凯伦,邹德安. 基于 LDA 模型的科研合作推荐研究[J]. 情报理论与实践,2015,38(9):79-85.
- [10] 施乾坤. 基于 LDA 模型的文本主题挖掘和文本静态可视化的研究[D]. 南宁:广西大学,2013.
- [11] 刘伟峰,杨爱兰. 基于 BIC 准则和 Gibbs 采样的有限混合模型无监督学习算法[J]. 电子学报,2011,39(3):134-139.
- [12] 石 晶,胡 明,石 鑫,等. 基于 LDA 模型的文本分割[J]. 计算机学报,2008,31(10):1865-1873.
- [13] 王 鹏,高 铨,陈晓美. 基于 LDA 模型的文本聚类研究[J]. 情报科学,2015(1):63-68.
- [14] 俞 辉. 基于 LSA 和 PLSA 的多文档自动文摘[J]. 计算机工程与科学,2009,31(9):108-111.
- [15] 牛 贺. 基于 PLSA 模型的推荐算法研究与实现[D]. 沈阳:东北大学,2012.

编辑 刘 冰