Applied Soft Computing 34 (2015) 178–193

Contents lists available at ScienceDirect

Applied Soft Computing

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

Emphasizing typing signature in keystroke dynamics using immune algorithms

Paulo Henrique Pisani[a,*], Ana Carolina Lorena[b]

[a] Universidade Federal do ABC (UFABC), Centro de Matemática, Computação e Cognição (CMCC), Av. dos Estados, 5001, Santo André, Brazil
[b] Universidade Federal de São Paulo (UNIFESP), Instituto de Ciência e Tecnologia (ICT), Rua Talim, 330, São José dos Campos, Brazil

ARTICLE INFO

Article history:
Received 24 January 2014
Received in revised form 11 January 2015
Accepted 10 May 2015
Available online 16 May 2015

Keywords:
One-class classification
Data pre-processing
Immune algorithms
Keystroke dynamics

ABSTRACT

Improved authentication mechanisms are needed to cope with the increased data exposure we face nowadays. Keystroke dynamics is a cost-effective alternative, which usually only requires a standard keyboard to acquire authentication data. Here, we focus on recognizing users by keystroke dynamics using immune algorithms, considering a one-class classification approach. In such a scenario, only samples from the legitimate user are available to generate the model of the user. Throughout the paper, we emphasize the importance of proper data understanding and pre-processing. We show that keystroke samples from the same user present similarities in what we call typing signature. A proposal to take advantage of this finding is discussed: the use of rank transformation. This transformation improved performance of classification algorithms tested here and it was decisive for some immune algorithms studied in our setting.

1. Introduction

The current technological scenario has brought a number of improved services to society, particularly owing to Internet-based applications. However, at the same time, this scenario has contributed to increase data exposure, giving a new momentum to concerns regarding identity theft. Thereby, there is a need to enhanced authentication mechanisms. A possible alternative is by the use of biometrics. In security area, biometrics tries to recognize users by physiological or behavioral features of the person.

There are several biometrics technologies currently available. This work focuses on keystroke dynamics, which studies ways to recognize users by their typing rhythm. This technology shows as being a promising alternative due to several reasons [1,2]. Firstly, it usually does not need any additional cost with hardware, as a common keyboard is enough to acquire keystroke data. Other biometric technologies, such as fingerprint or iris recognition, require a specific device to acquire biometric data. Secondly, keystroke dynamics recognition may be performed in background, while the user is typing an e-mail or entering a password. Consequently, day-to-day tasks are not disturbed, what may contribute to a better acceptability of the technology by the user.

A keystroke dynamics system should be able to distinguish a legitimate user from potential intruders, a classic binary classification setting in pattern recognition and machine learning. Nonetheless, collecting intruders data can be impractical in day-to-day use of computational systems. This makes a one-class classification setting more appropriate, where the user model is built using data from the legitimate user only.

Several algorithms have been applied for classifying users by keystroke dynamics, in both one-class and conventional two-class settings [3–5]. This paper focuses on immune algorithms, which attained good performance in some of our previous works [6–8].

In this paper we perform a deeper analysis of immune systems in the context of keystroke dynamics, analysing its performance under various aspects. The main goals are:

- Show that proper data understanding and preprocessing can be crucial in keystroke dynamics.
- Apply *rank transformation* in keystroke dynamics in order to improve recognition performance. This transformation can emphasize what is called here as *typing signature*.

Dealing with keystroke dynamics requires proper data understanding and preprocessing, as in the case of other areas [9,10]. Without it, classification algorithms may fail to reach optimal

* Corresponding author. Present address: Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação (ICMC), Av. Trabalhador São-carlense, 400, São Carlos, Brazil. Tel.: +55 163373 9700.
E-mail addresses: phpisani@icmc.usp.br (P.H. Pisani), aclorena@unifesp.br (A.C. Lorena).

http://dx.doi.org/10.1016/j.asoc.2015.05.008
1568-4946/

performance. This issue is even more crucial in a one-class setting, where one has to rely only on positive data for distinguishing both positive and negative data. In some cases, it may not be possible at all to perform data classification. To the best of our knowledge, there are not many papers dealing with data understanding and pre-processing in keystroke dynamics. This paper shows that some versions of immune systems are heavily affected by data pre-processing techniques, while others are more robust. This paper contributes by investigating how keystroke data behaves and showing a proposal to improve classifiers performance: the *rank transformation*.

In this work, we study the application of several immune algorithms along with an investigation on keystroke data. Several interesting results and conclusions from our research on keystroke dynamics in recent years, such as [8], [6] and [7], are discussed here in the context of data analysis. Next sections are organized as follows: in Section 2, we briefly introduce some previous work in keystroke dynamics; in Section 3, immune algorithms are briefly described in the context of keystroke dynamics; in Section 4, we present the concept of *typing signature* and how we can take advantage of it for improving classification performance; in Section 5, issues regarding data rescale in keystroke dynamics are discussed and we present our evaluation model; in Section 6, the results of applying *rank transformation* are discussed along with interesting findings regarding immune algorithms; in Section 7, we compare the performance of *rank transformation* to *decimal rescale*; and, in Section 8, we present our conclusions.

2. Keystroke dynamics

The area of keystroke dynamics has been studied for several years [5]. In order to identify the state of the art in keystroke dynamics, we conducted a quasi-systematic review [11]. Through the review, we identified main algorithms employed in this area and features extracted from keystroke data.

Table 1 shows a summary of main classification algorithms used in keystroke dynamics according to our recent review [11]. The number of users considered in each study and the error rate reported are also presented. Concerning the "Error rate" column, when there is a single value it stands for equal error rate (EER), otherwise it refers to false acceptance rate (FAR)/false rejection rate (FRR), respectively. FAR is the rate in which an intruder is wrongly accepted as being the legitimate user, while FRR is the rate in which the legitimate user is wrongly classified as an intruder. EER is the value in which FAR and FRR are equal. FAR, FRR and ERR are the most common evaluation measures employed in the area.

A common keyboard provides the instants when each key is pressed and released. Based on these data, a number of features

Table 1
Best performance achieved by classifiers (EER or FAR/FRR).

Classification algorithm	Users	Error rate
Random Forests [12]	53	1%/14%
Tree-based with Euclidean distance [13]	12	0%/3.47%
Gunetti and Piccardi: *R* measure [4]	205	0.005%/5%
AAMLP [14]	21	0%/0.25%
Gunetti and Picardi [15]	205	13%
SVM [16]	100	6.95%
Nearest neighbor [17]	51	9.96%
Hidden Markov Model [18]	20	3.6%
Bleha (with equalization) [19]	47	6.2%
Manhattan distance [20]	51	7.1%
GMM [1]	41	4.4%
Based on Gaussian distr. [21]	83	8.87%
Statistical [22]	100	6.9%

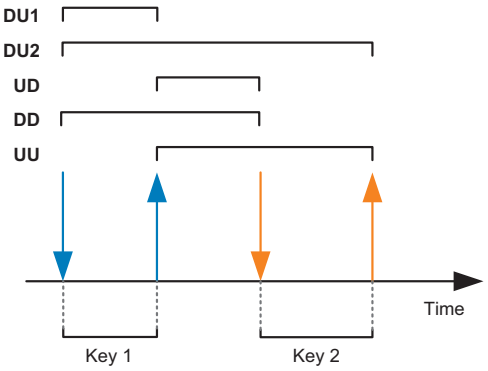

Fig. 1. Different extracted features in keystroke dynamics (adapted from [11]). Arrows down and up represent, respectively, when a key is pressed and released.

may be extracted in order to generate the feature vector, as shown in Fig. 1. All of these features are computed as time differences. For instance, the feature DU1 represented in Fig. 1 is the time difference between the instants in which a key is pressed and released. Some previous research also investigated the use of the pressure over the keys [23,22], however, it requires a specialized device.

Few studies performed a deep analysis of data in keystroke dynamics. However, some previous work deserve to be highlighted due to their different approaches to keystroke data. In [16] and [3], the authors studied the use of discretization over raw data. Each attribute in the feature vector was discretized into five values. The work of [15] applied a process of equalization in keystroke data. A comparison of different feature vectors in keystroke dynamics is done in [1].

Next section introduces immune algorithms. Afterwards, we discuss their use in keystroke dynamics.

3. Immune algorithms in keystroke dynamics

Artificial immune systems are computational systems inspired by the biological immune system and applied to solve problems [24]. These systems have been used in several applications related to pattern recognition, anomaly detection and optimization. This paper focuses on anomaly detection, which involves recognizing whether examples presented to the algorithm are legitimate or not. It is possible to draw a parallel between immune systems and the recognition of users by keystroke dynamics systems. Both need to generate a model of what is *normal* and be able to distinguish *abnormal* events (intruders) from this *normal* model.

According to a recent review on immune systems [25], negative selection algorithms are the most used in intrusion detection. This work focuses on these algorithms and also a strongly related class of immune algorithms called positive selection. Next sections present both immune algorithm classes: negative and positive selection.

3.1. Negative selection

Negative selection was introduced by *Forrest* [26]. As shown in Fig. 3, this algorithm is composed of two main phases: *censoring* (*training*) and *detecting* (*matching*). In the first phase, given a set of positive (self) examples, the algorithm generates random detectors and tests each of them against the available positive examples. Any detector which matches a positive example is discarded. This process is executed until a predefined amount of detectors is reached. The main idea is to cover the negative space with detectors (negative detectors), as shown in Fig. 2.

Afterwards, in the *detecting* phase, each example presented to the algorithm is tested against the detector set. If any detector