Ensemble of adaptive algorithms for keystroke dynamics

Brazilian Conference on Intelligent Systems, IV, 2015, Natal.
http://www.producao.usp.br/handle/BDPI/49976

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
Ensemble of Adaptive Algorithms for Keystroke Dynamics

Paulo Henrique Pisani
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Email: phpisani@icmc.usp.br

Ana Carolina Lorena
Universidade Federal de São Paulo
Instituto de Ciência e Tecnologia
Email: aclorena@unifesp.br

André C. P. L. F. de Carvalho
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Email: andre@icmc.usp.br

Abstract—Biometric systems have been applied to improve the security of several computational systems. These systems analyse physiological or behavioural features obtained from the users in order to perform authentication. Biometric features should ideally meet a number of requirements, including permanence. In biometrics, permanence means that the analysed biometric feature will not change over time. However, recent studies have shown that this is not the case for several biometric modalities. Adaptive biometric systems deal with this issue by adapting the user model over time. Some algorithms for adaptive biometrics have been investigated and compared in the literature. In machine learning, several studies show that the combination of individual techniques in ensembles may lead to more accurate and stable decision models. This paper investigates the usage of simple ensemble approaches by combining some current adaptive algorithms for biometrics. The main contributions of this study are:

- Proposal of a model to apply an ensemble of adaptive algorithms for biometrics;
- Study of the behaviour of the ensemble with adaptive algorithms in a data stream context, showing their predictive performance over time.

Biometrics in a data stream context implies in some challenges, as true labels are usually not available, even some time after the classification. In addition, the classification algorithms used here are all one-class, which receive only data from the positive/genuine user. In a recent paper [10], ensembles were used for adaptation in face recognition, however, the adaptation used samples provided by an operator.

In this paper, the ensemble approach is evaluated over three keystroke dynamics datasets, a behavioural biometric modality known to be subject to change over time [6]. This paper is organized in the following way: in Section II, related work is presented; in Section III, the ensemble approach is introduced, along with the proposals of ensemble for biometrics; in Section IV, the experimental setup is described; in Section V, the experimental results are shown, including an evaluation over time; and, in Section VI, the main conclusions are presented.

I. INTRODUCTION

Biometrics has been seen as a promising alternative to enhance the security of current authentication mechanisms. This is done by comparing physiological or behavioural features of the users, which is usually harder to be forged than a common combination of user and password. Biometric features should ideally meet several requirements, such as universality, collectability, distinctiveness and permanence [1]. Permanence refers to the fact that the biometric feature will not change over time. Nevertheless, some studies have shown that biometric features may change over time [2], [3]. As a consequence, the predictive performance of the biometric system may be negatively impaired for some users. In some studies, it has been named as template ageing in some cases.

Adaptive biometric systems have the goal of dealing with the issue of intra-class variability. This is done by adapting the user model/template over time so as to reduce the effect of template ageing [2], [4], [5]. Several papers have shown that adaptive methods can improve biometric prediction performance over time [3], [6], highlighting the importance of considering adaptation in a biometric system.

Several adaptive one-class algorithms have been proposed for biometric data streams [7], [8], [6]. Some of these algorithms are better when false match has to be reduced, while others have higher performance to reduce false non-match. Some studies have shown that the combination of individual techniques in ensembles may lead to more accurate and stable decision models. An example of ensemble in biometrics can be seen in [9]. Despite their good results in traditional, static datasets, the authors did not find studies using ensemble of one class classifiers for keystroke dynamics in a data stream context, where the data distribution may change over time. This paper investigates the use of simple ensemble approaches by combining some current adaptive algorithms for biometrics.

II. ADAPTIVE BIOMETRIC SYSTEMS FOR KEYSTROKE DYNAMICS

As mentioned earlier in this paper, adaptive biometric systems adapt the user model (sometimes referred to as user template) over time [4]. The issue of intra-class change has been observed for several biometric modalities, such as fingerprint and face recognition [2]. Given a set of user examples (labelled examples), adaptive biometric systems create a user model that is continuously adapted as new unlabelled examples are received during the system operation.

Keystroke dynamics is a biometric modality that attempts to recognize users by their typing rhythm [11]. Apart from the key itself, the keyboard provides the instants in which each
key is pressed and released. Based on this information, the recognition of users can be performed. Some recent studies have shown that the biometric features in keystroke dynamics may change over time [8], [6], indicating the need for the use of adaptive approaches.

However, there is a lack of public datasets suitable for studying keystroke dynamics over time. A dataset for this type of study has to meet some requirements, such as having several examples per user and such examples need to ideally be acquired in different sessions. For keystroke dynamics, we found three suitable datasets: CMU [12], GREYC [13] and GREYC-Web [14]. These are the datasets used in our experiments, more details regarding these datasets are presented in Section IV.

Some previous studies on model adaptation for keystroke dynamics can be found in: [7], [15], [8] and [6]. In [7], two simple methods based on the concept of galleries were discussed: growing window and moving window. Later, in [8], those approaches were further studied and a new method known as Double Parallel was proposed, which combines the ideas of growing and moving window in a new framework. Another work investigated adaptation in keystroke dynamics in a free text application [15]. The paper of [6] discusses the application of adaptive approaches based on the usage of detectors (Usage Control). Some of these adaptive algorithms are combined here in an ensemble approach.

Next section provides an introduction to ensembles and presents our proposal of ensemble application for biometrics using adaptive algorithms.

III. ENSEMBLE OF ADAPTIVE ALGORITHMS

Ensemble is defined as a method which induces several base classifiers and then perform classification based on the the outputs from these base classifiers. There are several ensemble approaches in the literature, such as majority voting, bagging, boosting [16]. This paper employs ensemble approaches related to fusion of label outputs: majority voting and stacking [17]. Next sections describe how we implemented each of them.

A. Majority Voting

![Fig. 1: Majority Voting (5 classifiers).](image)

In the majority voting (Fig. 1), several base classifiers receive the input example (query) and perform classification. If the majority of the base classifiers return positive, then the input example is classified as positive (genuine user) and, otherwise, as negative (impostor). All base algorithms used in this study are adaptive and, therefore, each base classifier is adapted/changed over time. The base classifiers are induced by the positive examples only. Hence, these are all one-class algorithms.

B. Stacking

![Fig. 2: Stacking.](image)

Another approach implemented here is stacking. In stacking, several base classifiers receive the input example (query) and perform classification, similarly to majority voting. However, in stacking, an additional classifier will receive the classification results from the base classifiers as input and then return the final classification result. This is illustrated in Fig. 2.

The base classifiers are induced by the positive examples only as they are all one-class algorithms. However, the stacking classifier is a two-class classifier which requires examples of both positive and negative examples produced by the base classification algorithms. This is a problem in a one-class scenario, because only positive (genuine) data is available for training.

In order to solve this problem in a biometrics scenario, we considered that the biometric system is able to access data from all known users at training time. This is feasible in biometrics in several cases. For instance, in an enterprise implementation, the biometric system is aware of all enrolled users. Based on this idea, we adopted the model shown in Fig. 3 to train the stacking classifier.

For each user, it uses the first half of the training examples to induce the base classifiers (one-class). Afterwards, a test is performed on the remaining training examples. In order to simulate impostor attacks, examples from other known users are used. As a result, a dataset of results obtained by the base classifiers and true labels are obtained. This dataset is then used to train the stacking classifier for each user. It is important to highlight that the training data is different for each user as the one-class classifiers are induced per user too. Note that although the base classifiers update their models over time (adaptive algorithms), the stacking classifier (non-adaptive algorithm), once induced, does not change over time. As we are considering a data stream context, the first examples of the user are used for training and the remaining ones are used on the same order they appear in the dataset. In this way, we also consider possible concept drift/behaviour change.

IV. EXPERIMENTAL SETUP

This section presents the experimental setup adopted in this paper.
A. Datasets and extracted features

The predictive performance of the investigated algorithms was assessed using three common datasets for keystroke dynamics:

- GREYC [13]: 100 users typed the expression “greyc laboratory” in at least five sessions, during two months. This dataset has more than 6,000 examples available considering all available users.
- CMU [12]: 51 users typed the password “tie5Roanl” plus the Enter key 400 times, during eight sessions. A total of 20,400 examples are part of this dataset.
- GREYC-Web [14]: 118 users contributed to this dataset, some of them for more than 1 year. This work uses the updated version, as available in the authors' website. Only the transcription of the login part is part of the current study, because it is closer to the setup of the other datasets used here (GREYC and CMU). Only the users with at least 100 examples were considered for this study, resulting in more than 7,000 examples from 35 users.

For this paper, the feature flight time type 1 [11] was extracted from the keystroke data. This feature is the time difference between the instants when a key is released and the next key is pressed. As shown in [18], this is one of the most used features in keystroke dynamics literature.

B. Evaluation methodology

In this study, the keystroke dynamics recognition task is seen as a one-class classification problem. Therefore, only examples from one class (genuine user examples) are used to induce the model (user template), except for the stacking ensemble described earlier in this paper. In the initial training phase, for each user, a classification model is induced by the learning algorithm using the first positive examples from the genuine user. Afterwards, in the test phase, a biometric data stream is generated (more details in the next section). The examples of the data stream are then presented to the algorithm, which classifies them and adapts the model. The examples in the data stream do not have a class label, thus, the algorithm does not know their true label. It is also important to highlight that, when generating a data stream for a given positive user, examples already used for training are not part of the data stream for that user.

The results reported in this paper are the average values considering all users, since the test is performed per user. Furthermore, due to the stochastic nature of the data stream generation (negative examples are interleaved randomly), all experiments are repeated 30 times.

C. Biometric data stream generation

The generation of the data streams is based on the user cross-validation methodology [19]. This methodology firstly divides the users into $N$ groups of similar size. $N$ assumed the value 5 in this paper, as in [19]. Based on these $N$ groups, $N$ test scenarios are evaluated. Each scenario considers the users in the $(N - 1)$ groups as the positive set and the remaining group as the negative only set.

A data stream is generated for each user in the positive set. The users in this set are considered the ones enrolled in the system (e.g. registered employees from a company). The generated data stream is formed by all test examples from the genuine user interleaved with examples from other users randomly chosen (impostors). The biometric stream has 70% of genuine examples and 30% of impostor examples, as previously considered in previous keystroke dynamics studies [8], [6], [19]. Among the 30% negative examples, there is a 50% chance of getting a negative example from the negative only set and a 50% chance of getting a negative example from the positive set. By using this procedure, we are able to simulate attacks from external users (negative only set) as well as internal attacks (positive set).
For all users (including impostors), the order in which the examples appear in the dataset is maintained. This is a key aspect, as it allows to verify possible concept drift/change in the way the user types on the keyboard through time.

D. Classification algorithms and parameters

Two static algorithms for keystroke dynamics are investigated in this study: Self-Detector [20] and M2005 [21]. Five previously proposed adaptive approaches for Self-Detector and one for M2005 are also part of our tests, as shown in Table I. The parameters of the algorithms assumed the same values as in [6]. For the ensemble, all five adaptive algorithms are used. In the case of the stacking, an additional classifier is used. The following classifiers were used in the stacking (Weka implementation [22]): Multilayer Perceptron, Decision Tree (J48), Random Forest and Naive Bayes.

TABLE I: Adaptive classification algorithms.

|-----------------------------|--------------------------------------|----------------------------------|-----------------------------------|------------------------------------|-----------------------------------|

V. EXPERIMENTAL RESULTS

This section presents the experimental results obtained with the individual classifiers and the ensembles.

A. Global results

Table II shows the overall predictive performance regarding false match rate (FMR), false non-match rate (FNMR) and accuracy rate (balanced version, due to data imbalance). FMR measures the rate of successful impostor attempts and FNMR measures the rate of genuine attempts wrongly rejected by the biometric system. They are both error rates, so the lower the value, the better is the performance.

According to the accuracy results, all adaptation methods performed better than the no adaptation case. As discussed in previous studies on adaptive biometrics [6], this is mainly due to the reduction in FNMR obtained by the user model being adapted over time. This suggests that the typing behaviour changes over time and that the adaptation of the user model has a key impact on the predictive performance.

The improvement on the accuracy between a static and an adaptive algorithm was higher in the CMU dataset. This may be because CMU data stream are, on average, longer. As a consequence, more changes can occur in the typing rhythm, enhancing the impact of applying an adaptive method. Regarding accuracy, the ensemble approaches obtained best results in the GREYC and GREYC-Web datasets, while in the CMU dataset the result was also among the best results. This illustrates that the usage of ensemble approaches in biometrics is a good strategy.

Comparing to the current adaptive approaches, the Ensemble (Voting) reached better accuracy in two datasets and obtained the second best performance in the CMU dataset. Hence, the ensemble performance was consistent among different datasets. The other adaptive algorithms (non-ensemble) obtained the best performance in one dataset, but not in the other two. It is illustrated by the fact that each dataset has a different best algorithm (considering the baselines). This consistent predictive performance may be an argument in favour of the ensemble, which, at the same time, implies in higher usage of computational resources.

Looking at the ensemble performance, we also observed that the stacking approaches had a tendency to increase FMR and decrease FNMR. Ensemble (Random Forest), for instance, was the best algorithm in terms of FNMR. This may be due to our method for training the stacking classifier. This classifier is induced with a test stream with 70% of positive examples, as the final test does, although only training examples are used. As a consequence, the stacking classifiers used more examples for the positive class during training. This may imply that they have a better fit to perform classification of positive cases, resulting in the improvement of FNMR, but at the cost of a higher FMR. The Ensemble (Voting) was not affected by this issue, as it does not need an additional stacking classifier.

| Algorithm FMR FNMR Acc (balanced) |
|-------------------------------|-----------------|------------------|
| Self-Detector (No adaptation) 0.090 (0.031) 0.165 (0.005) 0.872 (0.006) |
| Self-Detector (Sliding) 0.092 (0.111) 0.129 (0.004) 0.890 (0.005) |
| Self-Detector (Usage Control R) 0.092 (0.10) 0.140 (0.005) 0.884 (0.006) |
| Self-Detector (Usage Control S) 0.089 (0.10) 0.149 (0.005) 0.881 (0.006) |
| Self-Detector (Usage Control 2) 0.069 (0.09) 0.168 (0.006) 0.882 (0.006) |
| M2005 0.221 (0.19) 0.130 (0.003) 0.824 (0.009) |
| M2005 (I. Double Parallel) 0.210 (0.18) 0.092 (0.004) 0.849 (0.008) |
| Ensemble (Voting) 0.087 (0.010) 0.126 (0.005) 0.893 (0.006) |
| Ensemble (MLP) 0.181 (0.016) 0.054 (0.004) 0.882 (0.008) |
| Ensemble (Random Forest) 0.185 (0.16) 0.053 (0.004) 0.881 (0.008) |
| Ensemble (Naive Bayes) 0.116 (0.012) 0.094 (0.005) 0.895 (0.007) |
| Ensemble (Decision Tree) 0.184 (0.013) 0.066 (0.005) 0.875 (0.006) |

| Algorithm FMR FNMR Acc (balanced) |
|-------------------------------|-----------------|------------------|
| Self-Detector (No adaptation) 0.287 (0.032) 0.410 (0.016) 0.681 (0.009) |
| Self-Detector (Sliding) 0.291 (0.031) 0.211 (0.013) 0.749 (0.016) |
| Self-Detector (Usage Control R) 0.311 (0.030) 0.220 (0.013) 0.735 (0.015) |
| Self-Detector (Usage Control S) 0.213 (0.014) 0.275 (0.012) 0.756 (0.008) |
| Self-Detector (Usage Control 2) 0.143 (0.012) 0.323 (0.014) 0.767 (0.009) |
| M2005 0.273 (0.028) 0.451 (0.019) 0.638 (0.013) |
| M2005 (I. Double Parallel) 0.122 (0.011) 0.306 (0.008) 0.786 (0.006) |
| Ensemble (Voting) 0.208 (0.017) 0.239 (0.013) 0.736 (0.009) |
| Ensemble (MLP) 0.257 (0.039) 0.182 (0.018) 0.781 (0.012) |
| Ensemble (Random Forest) 0.283 (0.044) 0.168 (0.020) 0.775 (0.014) |
| Ensemble (Naive Bayes) 0.255 (0.025) 0.202 (0.010) 0.772 (0.015) |
| Ensemble (Decision Tree) 0.299 (0.043) 0.169 (0.016) 0.766 (0.014) |

| Algorithm FMR FNMR Acc (balanced) |
|-------------------------------|-----------------|------------------|
| Self-Detector (No adaptation) 0.066 (0.008) 0.141 (0.005) 0.896 (0.005) |
| Self-Detector (Sliding) 0.074 (0.011) 0.085 (0.004) 0.920 (0.007) |
| Self-Detector (Usage Control R) 0.069 (0.009) 0.096 (0.004) 0.922 (0.006) |
| Self-Detector (Usage Control S) 0.053 (0.007) 0.123 (0.005) 0.912 (0.005) |
| Self-Detector (Usage Control 2) 0.035 (0.007) 0.148 (0.010) 0.908 (0.007) |
| M2005 0.096 (0.013) 0.245 (0.016) 0.829 (0.008) |
| M2005 (I. Double Parallel) 0.095 (0.015) 0.131 (0.011) 0.887 (0.008) |
| Ensemble (Voting) 0.022 (0.001) 0.091 (0.004) 0.928 (0.005) |
| Ensemble (MLP) 0.126 (0.015) 0.052 (0.006) 0.911 (0.008) |
| Ensemble (Random Forest) 0.122 (0.026) 0.052 (0.004) 0.913 (0.012) |
| Ensemble (Naive Bayes) 0.087 (0.022) 0.067 (0.007) 0.923 (0.008) |
| Ensemble (Decision Tree) 0.121 (0.016) 0.063 (0.005) 0.908 (0.007) |

TABLE II: Global results for the three datasets (best results in bold and standard deviation within parenthesis).
Regarding the stacking ensemble approaches, we observed that the classifier used to combine the outputs may have an important impact on the results. One interesting result was observed for Naive Bayes, which assumes the independence of the input features [24]. This classification algorithm obtained the best stacking ensemble performance on two datasets: GREYC and GREYC-Web.

According to the Friedman statistical test [25] there are significant differences among the algorithms in terms of FMR and FNMR for \( p < 0.05 \). Regarding FNMR, Nemenyi post-hoc test [25] showed that Ensemble (Random Forest) is better than the other algorithms. In fact, this algorithm obtained the best FNMR on all datasets.

**B. FMR/FNMR over time**

The Figs. 4 and 5 show the FMR and FNMR over time, respectively. In order to do this, a window of size 50 was defined to measure these rates in steps of 10 examples. The average performance over all users of the first group division of user cross-validation is reported in Fig. 4 (similar tendencies were observed on other executions). These graphs show how the rates changed through the biometric data stream. Since the streams from CMU and GREYC-Web datasets are, on average, longer, only graphs for these datasets were plotted. Note that, in the GREYC-Web dataset, some users have more examples available than others. As these graphs show average results over all users, the analysis was limited just to the beginning of the stream for this dataset. All users have the same number of examples in CMU, hence the complete stream could be considered.

Firstly, regarding FNMR (Fig. 4), it is clear on the graphs that the static algorithms (Self-Detector and M2005) have a tendency to decrease performance over time. This suggests that the user behaviour has changed and the user model is not matching the newer typing patterns in later moments. On the other hand, the adaptive algorithms managed to obtain better values of FNMR. All Ensemble approaches reached similar performance, although Ensemble (Voting) obtained higher FNMR. This is related to the overall results seem in last section, in which all ensemble stacking approaches decreased FNMR while increasing FMR.

For FMR, the advantage of using adaptive algorithms is not clear. However, the main goal of the adaptive algorithms used here is to maintain the user model close to the user features (i.e. reducing false non-match). As a consequence, if these adaptive algorithms keep FMR stable over time, it is a good result. Among ensemble approaches, Ensemble (Voting) obtained the best FMR on both datasets.

**VI. Conclusion**

Several computational systems require secure authentication mechanisms and biometrics shows as a promising alternative to commonly used login and password combinations. However, biometric features may change over time. In order to deal with it, adaptive biometric systems have been proposed recently. This paper investigated the use of simple ensemble approaches for adaptive biometric systems. The experiments were conducted on three datasets for keystroke dynamics biometrics.

It is clear that ensemble approaches implies in a higher usage of computational resources, such as memory and processor time. However, according to the experiments performed in this study, the ensemble approaches implied in high predictive performance on all datasets. Ensemble (Voting), the simplest one, obtained accuracy better than current adaptive algorithms on two datasets and reached performance close to the best one in the other dataset. This consistent high performance may justify the use of an ensemble for adaptive biometrics.

This is a first investigation on the use of ensemble approaches on adaptive biometric systems. Some aspects of the current implementation can be further evaluated in future work. For instance, the rate of positive/negative examples in the data stream for training may be changed to improve the predictive performance of stacking approaches. In addition, other ensemble methods can also be studied.

**ACKNOWLEDGMENT**

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) for financial support.

**REFERENCES**


Fig. 4: False non-match rate (FNMR) over time.

Fig. 5: False match rate (FMR) over time.


