electronics

Article

Comparing Machine Learning Classifiers for Continuous Authentication on Mobile Devices by Keystroke Dynamics

Luis de-Marcos *, José-Javier Martínez-Herráiz, Javier Junquera-Sánchez, Carlos Cilleruelo and Carmen Pages-Arévalo

Departamento de Ciencias de la Computación, Escuela Politécnica Superior, Universidad de Alcalá, Ctra. Barcelona km 33.6, 28805 Alcalá de Henares, Madrid, Spain; josej.martinez@uah.es (J.-J.M.-H.); javier.junquera@uah.es (J.J.-S.); carlos.cilleruelo@uah.es (C.C.); carmina.pages@uah.es (C.P.-A.)

* Correspondence: luis.demarcos@uah.es

Abstract: Continuous authentication (CA) is the process to verify the user's identity regularly without their active participation. CA is becoming increasingly important in the mobile environment in which traditional one-time authentication methods are susceptible to attacks, and devices can be subject to loss or theft. The existing literature reports CA approaches using various input data from typing events, sensors, gestures, or other user interactions. However, there is significant diversity in the methodology and systems used, to the point that studies differ significantly in the features used, data acquisition, extraction, training, and evaluation. It is, therefore, difficult to establish a reliable basis to compare CA methods. In this study, keystroke mechanics of the public HMOG dataset were used to train seven different machine learning classifiers, including ensemble methods (RFC, ETC, and GBC), instance-based (k-NN), hyperplane optimization (SVM), decision trees (CART), and probabilistic methods (naïve Bayes). The results show that a small number of key events and measurements can be used to return predictions of user identity. Ensemble algorithms outperform others regarding the CA mobile keystroke classification problem, with GBC returning the best statistical results.

Keywords: authentication; authentication technology; mobile phone; data analytics; behavioral biometrics; typing dynamics; keystroke analysis; keystroke patterns

check for updates

Citation: de-Marcos, L.; Martínez-Herráiz, J.-J.; Junquera-Sánchez, J.; Cilleruelo, C.; Pages-Arévalo, C. Comparing Machine Learning Classifiers for Continuous Authentication on Mobile Devices by Keystroke Dynamics. *Electronics* **2021**, *10*, 1622. https://doi.org/10.3390/electronics10141622

Academic Editor: Manohar Das

Received: 20 May 2021
Accepted: 3 July 2021
Published: 7 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Mobile phones are rather pervasive today, with the prevalence of mobile phones, devices, and mobile communications increasing continuously. The sheer numbers and ubiquity of mobile devices tell about the necessity to establish methods that guarantee secure operation and communication. Authentication is the process of verifying identity of a system user. Authentication in mobile phones is commonly based on tokens like PIN, gesture patterns, passwords, and, more recently, on biometric-based techniques like fingerprint scans or facial recognition. However, attacks can bypass most authentication methods. PINs and passwords are susceptible to guessing or sniffing or more sophisticated methods like video side-channel attacks [1] or reflection reconstruction [2]. Smudge attacks can bypass patterns [3]. Even biometric systems are susceptible to spoofing [4]. Similar concerns arise for voice-based authentication [5]. Additionally, traditional authentication methods are a one-time process that is usually required to log in or unlock the device. Since mobile devices can also be taken without user permission (e.g., stolen), one-time authentication methods may result in unauthorized use even if the user authentication was initially legitimate.

Continuous authentication (CA) aims to mitigate all these shortcomings by running background processes that continuously monitor the user's interactions and characteristics to determine if the ongoing access is legitimate. Evidence on desktop computers suggests that even simple interactions with the keyboard feature unique individual traits [6]. Interaction with a mobile device is supposed to create a more detailed imprint because postural preferences [7] and other physiological traits like handgrip [8] come into play. CA is then particularly promising in a mobile scenario, but it also brings additional complications for the implementation since particularities of user interaction with mobile devices must be considered. Machine learning (ML) provides a set of classification algorithms that can tell apart legitimate user events from illegitimate ones, providing a backbone to build user models that can be used to implement CA. Since mobile devices and BYOD policies also bring new threats to organizations, threat analysis and threat intelligence efficiency rely on machine learning approaches' efficient application [9]. ML classifiers used in recent studies include random forests [10], neural networks [11], or even deep learning [12]. However, to the best of our knowledge, there is little evidence comparing and reporting classifiers' performance under the same conditions ([13] is a notable exception). The existing studies differ significantly in critical factors, such as the features selected, their extraction or normalization. The evidence on intrusion detection suggests that feature selection and extraction influence efficiency and effectiveness [14]. The studies also usually focus on tuning the models to beat a given accuracy benchmark for the classifier and dataset under scrutiny. Further, these studies usually compare their results with a small subset of different, and sometimes unrelated, results reported in other studies.

The most common interaction method monitored for CA is keyboard input. Typing determines a unique pattern that has been investigated for traditional keyboards [15] and different variants of mobile keyboards [16]. Standard interaction methods are preferred for CA because they rely on the metrics gathered unobtrusively during regular sessions. Although biometric authentication methods like facial recognition can be used, they usually present several practical problems that designers have to face. Firstly, biometric data demand specific protection and privacy features in systems that deal with them, although specific legal requirements vary depending on the geographical location. Secondly, biometric authentication methods depend on the availability of resources (e.g., cameras) to capture data. The device may deny access resulting in interruptions of the user's regular interaction similarly to other token-based authentication methods. In the mobile environment, modern smartphones provide additional input sensors and can capture user gestures. All these can be combined with keystrokes to provide a lot of data of user interactions that can be used for CA. However, this research body tends to produce ad-hoc solutions that rely on a complex operational process with multiple stages (data gathering, feature extraction, and decision-making) difficult to extend or implement in broader contexts.

This research contributes to knowledge by:

- presenting the results of training and comparing ML CA models based on keystroke mechanics that use substantially fewer features than the current state-of-the-art models but nonetheless offer comparable results.
- showing that a small number of key events and metrics return accurate predictions of the user's identity.

The results are also relevant for practitioners and the broader access control community since ML CA models can be used to implement or feed mobile agents that can respond to incidents. In this way, communication between different agents (e.g., client–server) will be more efficient. Further, our approach also results in user CA models that can be efficiently built, maintained, and updated.

The rest of the paper is structured as follows. Section 2 presents the literature review of CA and keystroke mechanics. Section 3 presents the methodology of the study. The results are presented in Section 4. The paper closes with the discussion and conclusions.

2. Literature Review

Continuous authentication is the process of determining the legitimacy of the user's identity without their active participation. CA contrasts with traditional authentication that usually relies on system credentials provided once to identify the user. CA systems typically run as a background process that gathers information about physical or behavioral properties to determine the identity. The first and most popular method of CA is to use keyboard interactions [17]. Measurements of keypresses like the down–up or up–down time can be used to define individual patterns. They can be taken for every single key event (usually called monograph features) or for a sequence of two (digraph features) or more keys. The latter facilitates determining the latency of presses and releases between different events. CA models for keystroke dynamics achieve high accuracy with a low false acceptance ratio even for free-text input [18]. The body of study of these techniques is usually called keystroke dynamics or keystroke behavioral biometrics. A systematic literature review of keystroke dynamics can be found in [19].

Given the existing body of research on PC-based keystroke dynamics, it is not surprising that initial works on CA for mobile devices also focused on keyboard events. Seminal works on mobile phone CA focused on keystroke mechanics with a hardware keypad included in the first generations of handset devices [11]. However, with the spread of touchscreen mobile devices, this body of work has been adapted to virtual keyboards as they became commonly available in smartphones. Teh et al. [16] presented a literature review of touch dynamics biometrics for virtual keyboards. They divided the operational process into three stages: data acquisition, feature extraction, and decision-making. The decision-making techniques reported in the literature are probabilistic modeling, cluster analysis, decision trees, support vector machines, neural networks, distance, and other statistical methods.

Further, smartphones provide two additional elements that can be used to capture additional data to feed CA models and processes. Firstly, they include a set of sensors (e.g., an accelerometer, a gyroscope), the input values whereof can be captured at any given moment or event. Second, touchscreens provide the capability to capture user gestures. The input associated with these interactions (e.g., position or pressure) can also be monitored during the gesture. All this additional input provided the ground for the third generation of mobile CA that takes advantage of sensor and gesture data. Sensor-enhanced keystroke mechanics improve gesture-based authentication and traditional keystroke mechanics [20]. Shunwandy et al. presented a literature review on sensor-based authentication, although they focused on healthcare [21], a special sensitive domain for authentication [22]. Experimentation with touch features shows that they provide reliable short-term CA predictions which can be effectively combined with other long-term authentication methods [23]. Hand movement, orientation, and grasp (HMOG) is a set of behavioral biometrics for smartphones introduced by Sitová et al. [24]. It includes two types of features (resistance and stability) that can be used on their own or combined with others (taps, keystrokes, and sensors) for CA. Sitova et al. reported that HMOG outperforms individual sensors. The best results come, however, when HMOG is augmented with tap and keystroke features. Their results also show that HMOG features are particularly suited for CA during walking sessions.

Smith-Creasey and Rajarajan [10] presented a gesture-typing method on mobile phones that can authenticate users for each word. Gesture typing is a different input method in which users press and slide their finger between the characters that form the word that they want to type. Their scheme considers unique aspects of gesture typing, such as redirections and pauses. They reported an error rate of 3.5% for a single-word gesture and 0.8% for three-word gestures. Although this method yields the best results reported in the literature, it relies on an unusual input method. It also requires extracting a significant number of features from gestures and subgestures and undertaking normalization and fusion techniques with extracted data.

However, the current literature focuses on improving CA methods' accuracy by applying a multistage process that usually includes data gathering, feature extraction, normalization, model building, and testing. This process makes it difficult to compare classifiers to the extent that it is questionable whether such complexity presents a substantial improvement. To our knowledge, the only study to approach mobile CA from a comparative perspective has been carried out by Serwadda et al. [13] who reported a dataset and a controlled experiment to compare the performance of ten classifiers for touch gestures on smartphones. They concluded that logistic regression outperforms other classifiers for vertical strokes. SVM, random forests, and logistic regression returned similar results for horizontal strokes, although they outperformed all the other methods studied.

Since current research relies on a myriad of input data and complex modeling to continuously authenticate mobile phone users, this study sets out to study the feasibility of using lighter CA agents based on metrics from a single input or sensor. This approach results in more scalable CA systems than the current state-of-the-art mobile CA methods, providing acceptable accuracy levels for user prediction. Further, this study also aims to build authentication models that are based on one or a short sequence of events using ML algorithms, and it also compares the accuracy of different ML classifiers.

3. Methodology

For this research, we trained and tested seven ML CA models that can return predictions of user identities for each single keypress event recorded by the soft keyboard of mobile phones. The following subsections report the measurements used in this study, the dataset, and the ML classifiers and metrics used to compare the performance of CA models.

3.1. Measurements of Keystroke Dynamics

The following keystroke mechanics metrics were considered for the mobile CA agent considered for this study (Figure 1): pressingTime, timeReleaseNextPress, and timeBetweenPress. PressingTime is the key hold latency between the press and the release of a given key of the soft keyboard. This measurement is also called down–up time, and it is a key hold feature for each key pressed. TimeReleaseNextPress (up–down time) is the time between the key's release and the following keypress. TimeBetweenPress (down–down time) is the time between the press of a key and the next keypress. Down–down time and up–down time are considered digraph features since they consider two consecutive keypresses. Measurements of single keypresses, like pressingTime, are called unigraph features. All the measurements were taken in milliseconds for each key pressed. Additionally, it is also possible to record the key code of the present key (keyCode) and the key code of the next key pressed (nextKeyCode).

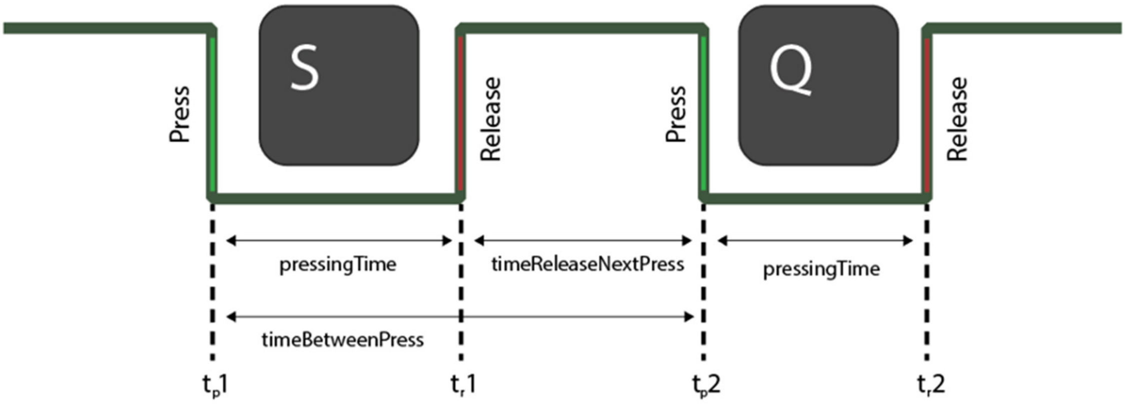

Figure 1. Keystroke mechanics of this study.

3.2. Dataset

The data used in this study come from a public HMOG dataset (http://www.cs.wm.edu/~qyang/hmog.html; accessed on 18 December 2019) [24,25]. A HMOG dataset records interactions of 100 users during 24 sessions (eight reading sessions, eight writing sessions, and eight map navigation sessions). It includes sensor information, touches, gestures, and keypresses on the virtual keyboard. The HMOG dataset recorded raw data for each keypress event, including the timestamp, type (down or up), and key code. For this study, the keypress event information of all the eight writing sessions of the HMOG dataset was transformed using a Python script that computed pressingTime, timeBetweenPress, and timeReleaseNextPress based on the timestamps and types of keypresses reported in the dataset. The transformed dataset used in this study includes 712,418 keypress events for 100 users (from 4306 to 11,917 events). Descriptive statistics of the variables for all the users are presented in Table 1.

Table 1. Descriptive statistics of measurements for the transformed dataset (*n* = 712,418).

Measurement	Mean	SE of the Mean	SD	Median
pressingTime	92.82	0.03	28.26	91
timeBetweenPress	455.06	0.60	504.10	309
timeReleaseNextPress	362.24	0.70	504.82	212

Datasets for training ML models were subsequently built for each participant. To do so, each dataset included the user's keystroke events and a random sample of events from other users. Events from the user were labeled as authorized or legitimate (positives), and random events from other users were labeled as unauthorized or malicious (negatives). Each dataset included approximately 50% authorized events and 50% unauthorized events. Table 2 presents the dataset's headers and a few data entries that provide an example to show its structure.

Table 2. Structure of the transformed dataset: headers and a sample of data entries.

Key Code	Pressing Time	Time between Press	Time Release Next Press	Next Key Code	Authorized
32	112	666	554	107	1
107	110	104	403	110	1
110	82	517	435	8	1
8	103	1270	1157	99	1
99	85	935	850	115	0
115	108	17	–91	101	0
101	140	237	97	32	0

3.3. Machine Learning Classifiers

ML models for continuous authentication of users can be implemented using ML classifiers. The second objective of this study is to compare ML algorithms for mobile CA using keystroke dynamics. Since the CA problem is a classification problem that aims to tell apart authorized events from unauthorized events, we focused on classification-supervised algorithms. The algorithms tested in this study included a variety that covers the most common and popular categories of classifiers. Ensemble methods are composed of weaker models independently trained and combined to make an overall prediction. It is a class of algorithms that is rather popular nowadays. This study included three ensemble algorithms: random forest classifier (RFC) [26], extra trees classifier (ETC) [27], and gradient boosting classifier (GBC) [28]. Instance-based algorithms make decisions based on other known instances of the problem considered important or representative, typically using a similarity measure. This class was represented in this study by the k-nearest neighbors (k-NN) algorithm. Hyperplane methods like support vector machines (SVM) compute a mathematical model using a hyperplane that consists of a set of decision boundaries used to classify datapoints. The points are then classified according to the side of the hyperplane in which they fall. Decision tree algorithms build a model of decisions based on the values of attributes in the data. Classification and regression tree (CART) was included in this study. Finally, naïve Bayes was included to represent Bayesian algorithms. For a detailed description of classification algorithms see [29]. An ML model for each of these algorithms was implemented in Python using the scikit-learn module [30].

Crossvalidation with five folds was used to train and test the classifiers. We used the standard target ML metrics to compare ML algorithms' performance: accuracy, precision, recall, and F1. Accuracy is the ratio between correct predictions and the size of the dataset. Precision measures how many of the predicted positives are true positives. Recall returns how many of the actual positives (true positives + false negatives) the model can predict. Precision should be the target metric when the costs of a false positive are high, while recall is preferred when a false negative is high. F1 represents a balance between precision and recall used when there is an uneven class distribution in the dataset. A false positive may entail a higher cost for CA since it means that an unauthorized person uses the device. A false negative would mean blocking the device or session of authorized users requiring that they authenticate again. The receiver operating characteristic (ROC) curve was also produced for each participant and all classifiers. The area under the curve (AUC) is a summary statistic of the ROC curve representing the probability of ranking a randomly chosen positive instance higher than a randomly chosen negative one. It is representative of how much the model is capable of distinguishing between positives and negatives. Since ML classifiers for the CA problem are binary, we also included the Matthews correlation coefficient (MCC). MCC returns a value between -1 and 1, representing the correlation between the true and the predicted classes. Accuracy is sensitive to class imbalance while the other standard metrics (precision, recall, F1) are asymmetric. MCC is a more reliable statistical metric that produces a high score only if both classes are predicted well, even if one class is disproportionately overrepresented [31].

4. Results

Table 3 shows the average results returned by the different ML classifiers for the CA problem for the 100 users extracted from the HMOG dataset. The results of each metric are also presented graphically in Figure 2. Ensemble algorithms (RFC, ETC, GBC) performed better, with an average of over 70% for most target metrics. The results show high variability between participants, with accuracy ranging between 0.58 and 0.91 across users. Ensemble methods are followed by k-NN, which outperforms SVM. It returns an average accuracy of 0.65, although variability ranges substantially (between 0.56 and 0.89) for ensemble algorithms. SVM returns the worst performance of all classifiers with an average accuracy of 0.59 (from 0.51 to 0.70). Four different kernels were tested (linear, sigmoid, polynomial, and RBF). The radial basis function (RBF) returns substantially better results than others, and it is used for testing and comparison. Naïve Bayes performs similarly to k-NN and CART, and it also shows a considerable variability among users with values ranging between 0.54 and 0.84. The decision tree classifier implementing the CART algorithm returns the second-lowest accuracy measure with an average of 0.63 (from 0.55 to 0.86).

Table 3. Results of ML classifiers for the CA problem. Average of target metrics.

Classifier	**Accuracy**		**Precision**		**Recall**		**F1**		**AUC**		**MCC**	
-	**M**	**SD**	**M**	**SD**	**M**	**SD**	**M**	**SD**	**M**	**SD**	**M**	**SD**
RFC	0.68	0.06	0.71	0.06	0.76	0.07	0.73	0.06	0.72	0.07	0.59	0.12
ETC	0.67	0.06	0.70	0.06	0.74	0.07	0.72	0.06	0.71	0.07	0.57	0.12
GBC	0.71	0.05	0.71	0.06	0.83	0.06	0.76	0.05	0.74	0.07	0.63	0.11
k-NN	0.65	0.06	0.68	0.06	0.74	0.07	0.71	0.06	0.67	0.07	0.48	0.13
SVM	0.59	0.04	0.60	0.05	0.68	0.15	0.61	0.12	0.60	0.05	0.19	0.06
Naïve Bayes	0.64	0.06	0.66	0.07	0.72	0.14	0.72	0.08	0.67	0.08	0.45	0.13
CART	0.63	0.05	0.68	0.06	0.66	0.06	0.67	0.06	0.61	0.06	0.41	0.11

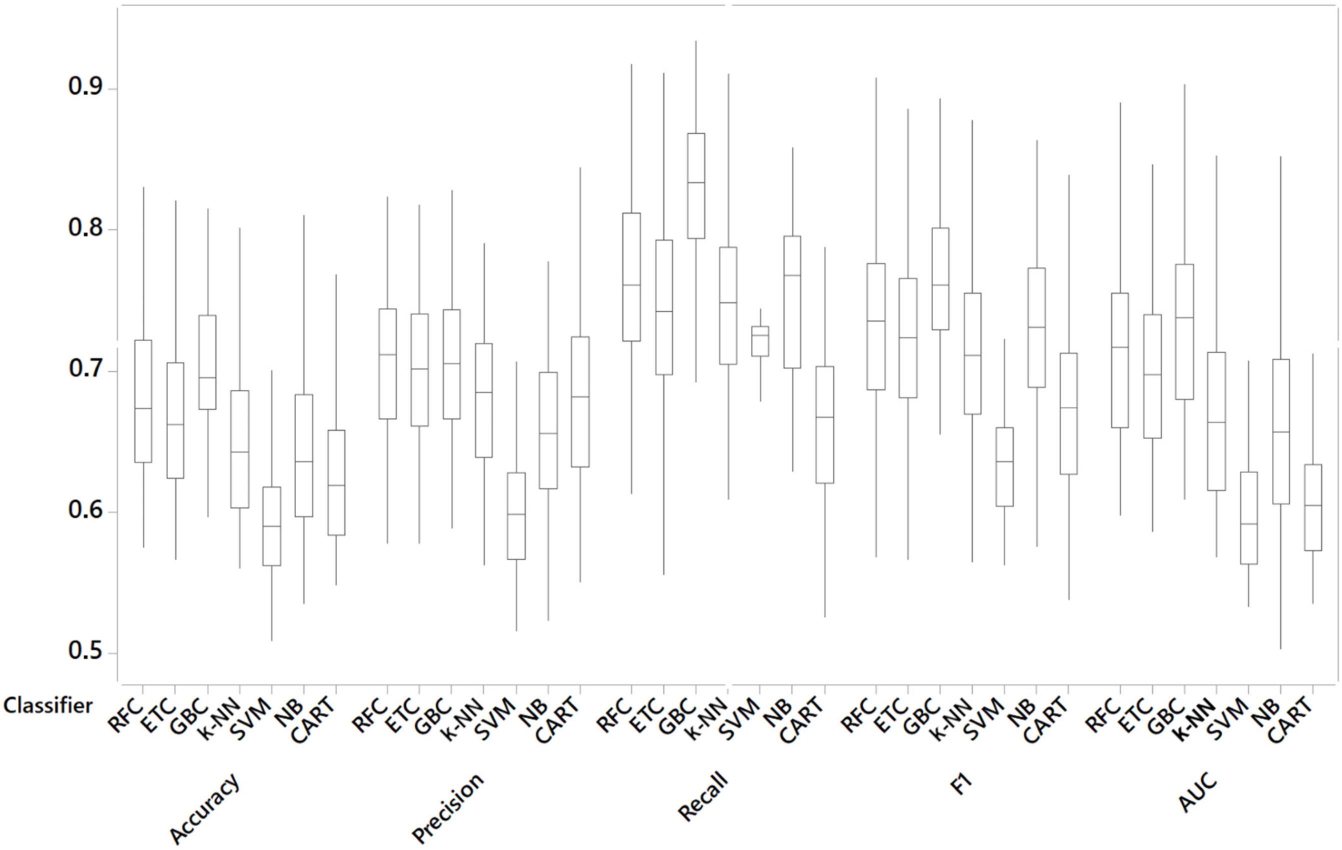

Figure 2. Boxplot of metrics (accuracy, precision, recall, F1, AUC) of ML classifiers for the CA problem.

We run an analysis of variance (ANOVA) to statistically compare the differences between the 100 samples for each classifier. The results showed that the differences were significant across all the metrics: accuracy (F = 45.41, p < 0.001), precision (F = 42.81, p < 0.001), recall (F = 31.90, p < 0.001), F1 (F = 45.99, p < 0.001), AUC (F = 57.41, p < 0.001), and MCC (F = 169.80, p < 0.001). The margin of error was 0.01 for a 95% confidence interval of the means for accuracy, precision, F1, and AUC. The margin of error for recall and MCC was 0.02. Tukey's pairwise comparisons showed that GBC outperformed all the other methods statistically for accuracy and recall. For F1, AUC, and MCC, there were no statistical differences between GBC and RFC, although GBC outperformed all the other classifiers. For precision, there were no statistical differences between the three ensemble methods. Figure 3 presents the confidence intervals for accuracy, showing the differences graphically. GBC performed better, while several other groupings are also observed. Overlapping intervals (for each pair of classifiers) in Figure 3 mean that there were no differences. Non-overlapping intervals mean that there were statistical differences between the two methods.

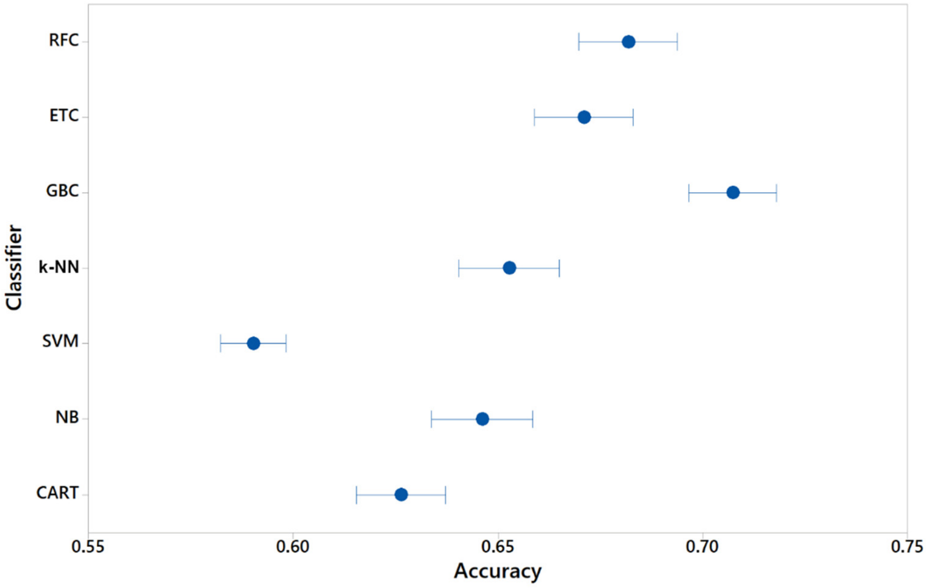

Figure 3. Interval plot of accuracy for all the classifiers (95% confidence interval of the mean). Overlapping between pairs of intervals means that there are no statistical differences.

All the classifiers returned similar scores for all the metrics suggesting that they were not biased towards predicting more false positives than false negatives (or vice versa). GBC and k-NN returned substantially higher values for recall when compared with other metrics. As the recall metric is preferred for the mobile CA problem using keystroke dynamics, the results suggested that GBC was better for the given dataset. The other ensemble classifiers (RFC, ETC) returned a similar result, representing a feasible option to implement ML CA models. As for MCC (Figure 4), the results showed a strong positive correlation (MCC > 0.5) for the three ensemble classifiers. All the other classifiers also returned a moderate positive correlation except SVM that showed no correlation. Figure 5 presents the ROC curve of all the classifiers and the AUC values of an arbitrary user. We can observe that GBC performs better, followed by the other two ensemble classifiers, which perform similarly. Next comes k-NN while CART, naïve Bayes, and SVM perform substantially worse in terms of distinguishing between positives and negatives.

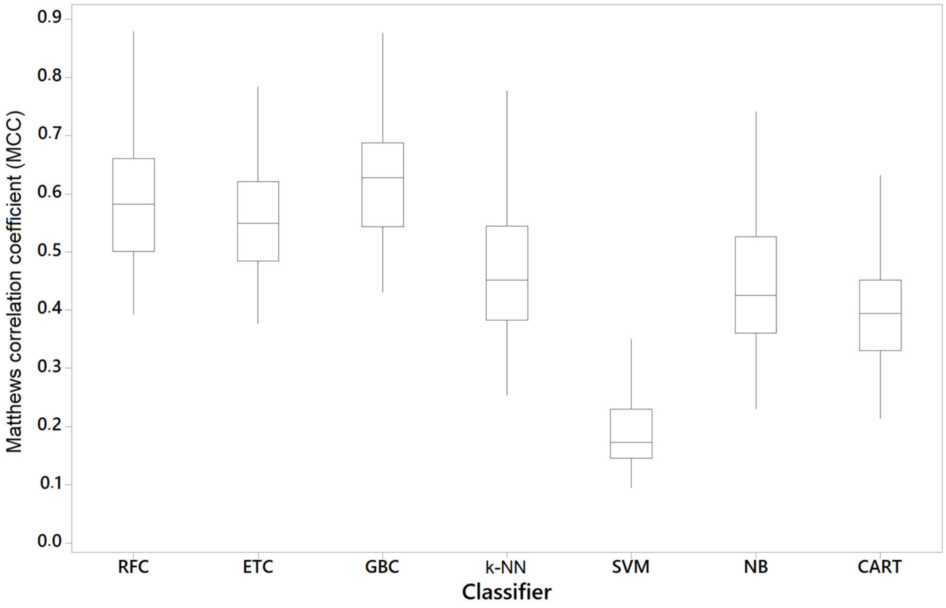

Figure 4. Boxplot of Matthews correlation coefficient of ML classifiers for the CA problem.

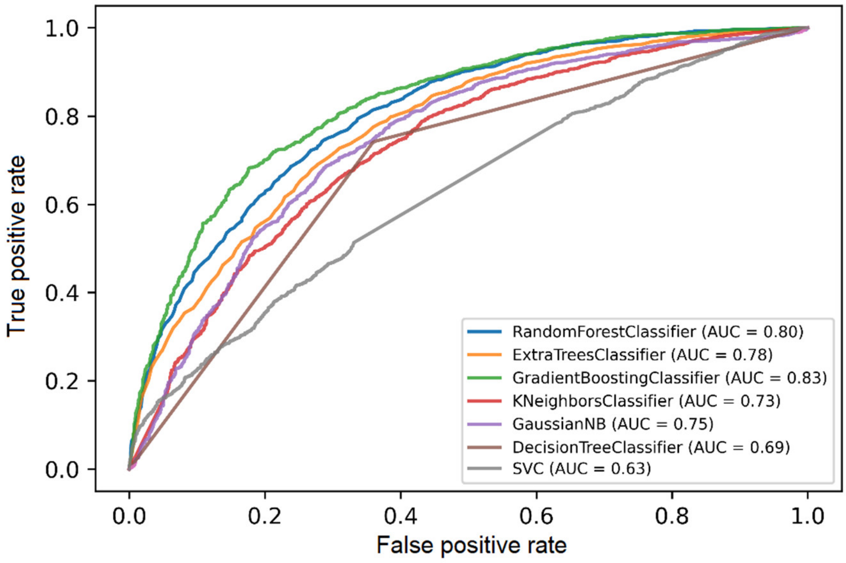

Figure 5. ROC curves of the different classifiers for an arbitrary participant.

Ensemble classifiers (RFC, ETC, and GBC) and the tree classifier (CART) also return the importance of each feature. The most important feature was found to be pressingTime, ranging between 28% and 50% on average for all the classifiers. It was followed by timeReleaseNextPress (14–23%), timeBetweenPress (12–22%), keyCode (12–15%), and nextKeyCode (9–14%). This finding suggests that the three keystroke measurements play a role, each contributing to the final prediction. Previous studies on keystroke mechanics on desktop computers also showed that pressingTime plays a more significant role, which our study also supports for mobile phones. The keys pressed and their sequence in a digraph are the least important features to determine the class to which each event belongs.

The results may suggest that the prediction is not very accurate since even for the best ML classifiers, around 29% of the cases are incorrectly classified. However, every single event (i.e., key pressed) produces a prediction. Therefore, a few individual predictions can be combined to produce a more reliable result, thus mitigating the number of false positives and false negatives. Indeed, current literature suggests using a combination of events for mobile phones or several keypresses (usually in the form of a word or short text) for keystroke dynamics authentication [11]. For this study, the probability of having two false negatives (or positives) in a row is around 0.084, and the probability of having four consecutive false negatives (or positives) is under 0.007 for the best classifier (GBC). Therefore, in the final implementation, a mobile CA agent should respond, e.g., block the device, only if several successive unauthorized predictions or a high percentage are found in recent events. The number of events to consider can be a parameter that can be fine-tuned for each user.

5. Discussion

This paper implemented and compared different ML agent models for CA in mobile environments. Although specific, scalable architectures have been presented [32], to the best of our knowledge, this is the first implementation and testing with specific models for keystroke dynamics using different classifiers on the same dataset. The results suggest that ensemble classifiers (RFC, ETC, and GBC) work better for the problem at hand than instance-based algorithms (k-NN), hyperplane methods (SVM), Bayesian models (naïve Bayes), and decision trees (CART). Notably, GBC outperformed all the other classifiers with statistically significant differences. Ensemble decision classifiers use multiple learning algorithms, typically multiple decision trees, reporting the class that is the mode of all (RFC, ETC). GBC also combines several weak decision trees but using gradient boosting. CART is based on a single strong decision tree classifier that maximizes the information gain at each node. Therefore, we argue that the combination of several weak decision trees works better for the CA problem and the sample given. Individual decision trees like CART usually face overfitting problems resulting in poorer results for the evaluation set, as we could observe here.

All in all, the difference was around 7–8% for all the target metrics when CART was compared with GBC. Instance-based algorithms (k-NN) make a decision based on individual instances like the majority of a given number of neighbors; k-NN is comparable with ETC and only performs worse than GBC and ETC, suggesting that distance to neighbors can also be a good estimator of the legitimacy of individual key events. This may be particularly useful in environments that require fast response times or have to work with limited training samples since k-NN algorithms are easy to implement, fast, and require fewer data. The drawback of a potential 5% decrease in accuracy can be mitigated by increasing the number of keypresses necessary to make a decision by a mobile agent. Hyperplane methods (SVM) compute the mathematical model that separates most instances when represented as individual points in a hyperspace. SVM returns poor results for the CA problem suggesting that this problem is difficult to model using hyperplanes to classify instances. Our study considered one probabilistic classifier, and we can see that it performed worse than all the ensemble classifiers, although its results were comparable to k-NN and CART. Since Bayesian classifiers consider features to be independent, we argue that this may not be the case for the CA problem using the keystroke measures considered in this study.

When comparing the three ensemble classifiers, we found that GBC outperformed the two others, while no significant differences were found between RFC and ETC. Previous evidence suggests that GBC usually outperforms random forests for classification problems, and our findings suggest that this is also the case for the mobile CA problem with keystroke mechanics. All in all, differences in target metrics were around 4%, and the statistical difference reported here may be caused by sample size. This suggests that gradient boosting with weak decision trees provides a small benefit over ensembles of tree classifiers that return the mode of the forest. The reason may lie in the nature of data, particularly with low dimensionality, since boosting algorithms usually benefit from a large number of features. Differences for the target metrics studied between random forests (RFC) and extra trees (ETC) were marginal and not significant. Extra trees differ from random forests in two aspects. Firstly, random forests select the optimal cutpoint in the splitting process of the tree for each feature, while in extra trees, the point is randomly selected from a uniform distribution of the feature's range. Secondly, ETC uses the whole learning sample to train each tree while RFC uses a sample bootstrap. As results do not return significant differences between RFC and ETC, we argue that the random nature of the splitting point and the set used for training each tree do not yield a substantial benefit in terms of classifier performance. The CA problem and dataset gathered are not affected by the variations coming from the implementation of different ensemble algorithms.

As for the previous studies comparing classifiers, Serwadda et al. found that logistic regression, SVM, and random forest outperformed other classifiers for touch-based CA. They reported error rates ranging between 10% and 21% under different conditions: device orientation (portrait, landscape) and stroke orientation (vertical, horizontal) [13]. Their results contrast with our findings, which suggest that ensemble algorithms perform better, followed by k-NN. In our tests, SVM performed poorly. In Serwadda's results, decision trees and k-NN performed poorly. We did not train a logistic regression classifier. Their approach sampled touch-based CA events at regular intervals, extracting and deriving features from the data acquired, which were subsequently used to train the models. Keystroke dynamics produces data from events that can feed training algorithms directly or after a relatively simple extraction. Such differences can explain why statistical methods work better for touch-based strokes while k-NN works better for keystroke dynamics. Ensemble methods performed well in both cases (although Serwadda et al. only reported the random forest), returning promising results to guide future investigations.

The results also suggest that a small number of keystroke measurements is sufficient to provide accurate predictions of user identity. Our results are comparable to the state-of-the-art studies on PC keystroke dynamics [18]. Clarke and Furnell [11] reported error rates of 8% and 9% for inputs of eleven digits and four digits on mobile phones' hardware keypads. Studies on gesture typing return error rates around 3.5% for one word and under 1% for three words [10]. Our findings also suggest that similar results can be obtained using the soft keyboard's measurements with only a few characters when the user model is trained. This also mitigates the possible effects that typing bursts may have for mobile CA. CA mobile agents can make decisions and take actions even if user interaction takes the form of short bursts. As for training, agents may gather data during users' regular interactions independently of their typing form to get enough interactions to train their models. The effect of fatigue and typing bursts is also an interesting line of future development for mobile CA.

Several previous studies about keystroke dynamics consider additional measurements like the time between releases (also called up–up time) or the total time between the press of the first key and the release of the second key [16]. However, these are just linear combinations of pressingTime and timeReleaseNextPress. We tested this and other linear combinations of the measurements used in this study and did not find any substantial difference. This result suggests that the measurements selected are sufficient to profile most users and that ML methods can handle possible collinearity between variables, thus not benefiting from features that are just linear combinations of others for the CA problem. A possible issue, however, is the high variability returned by all the classifiers. It suggests that there are users for whom no accurate fingerprint can be learned with a given method, as reported in previous studies [33]. This stresses the necessity of combining several inputs and classification methods to authenticate the majority of participants successfully.

The results of the body of work that uses gestures and/or sensors for CA in mobile phones are difficult to compare with our findings given the fundamental differences in the procedure and features used. However, HMOG [24] also considers key events, so stressing the differences can provide additional insights pointing to the benefits of combining both. HMOG uses fewer key events, focusing only on features of single key events (unigraph). Our study uses the HMOG dataset extracting the digraph features that represent the interaction between a keypress and the following. Provided that the lowest error rates in the original HMOG study were found when HMOG was combined with key and tap events, feeding CA models with additional keypress features may improve the accuracy.

A major drawback of our study is that decisions were based on a single key event resulting in relatively low performance, although comparable with the current state of the art as discussed in this section. In the final part of the results, we suggested that several decisions can be combined to get a better insight. Here, we provide an outline of a workable application. Practical implementation can adopt a voting system that can generate a combined trust value and a decision for a given user. The final CA system can use an API that trains and implements several ML models. Each model can still employ different user measurements (e.g., keystroke dynamics) and have a different weight in the final decision. When the CA system collects sufficient information, each model generates a trust value. The trust value can be based on any ML metrics for a sequence of events or a combination of them. Then, the voting takes place. The weight of each vote should be based on the accuracy of the model. For instance, a model with an accuracy of 96% will have more weight on the final decision and trust than a model with an accuracy of 90%. For each user, given the trust (T_i) and the accuracy (P_i), we can use the following weight sum to compute the final trust:

$$T = \frac{\sum_i (T_i \times P_i)}{\sum_i P_i}$$

The voting system outlined here is an ensemble system. Systems based on complementary methods already showed their potential in practical applications, like recommender systems [34], which also describe how to evaluate them.

This study presents other limitations. The participants' representativity and sample size may be a threat to validity since data come from a public dataset of 100 volunteers over eight writing sessions. The original HMOG dataset did not provide substantial information of participants besides gender. We could not assess the effect of possible unbalances in the sample like age, language, or experience with mobile phones, limiting the generalization of our findings. The reduced number of sessions is somehow mitigated by a large number of participants and of events per participant, which facilitate a good statistical representation. Representativity of the sessions is also a limitation since the creators of the HMOG dataset designed these to represent everyday interactions. Research on attack detection shows that unknown attacks are difficult to learn from [35], and as CA usually models impostors as the action of others, CA systems may respond poorly to new attack vectors. Several studies also analyzed the environmental conditions of the interaction, such as posture (e.g., walking, sitting), which our study did not address. Other studies also included device orientation (portrait, landscape), which our study did not analyze either since we considered all the key events of the writing sessions of the original HMOG dataset as equally representative of users' typing interactions. Besides all these limitations, this study establishes the experimental conditions required to make the results of machine learning classifiers comparable, establishing a testbench that can guide future research and practitioners of mobile CA systems.

6. Conclusions

This paper presented an agent model that facilitates the integration and development of CA in mobile devices. Seven different classifiers were then trained and tested using keystroke dynamics captured from mobile devices' soft keyboard events from the HMOG dataset. The results show that all the digraph features used in this study (down–up, up–down, and down–down time) were relevant for the CA classification problem. Ensemble algorithms (RFC, ETC, GBC) performed better, with an average accuracy of around 0.70 for every single key event. GBC outperformed all the other classifiers, and the differences were statistically significant. Naïve Bayes and k-NN returned an accuracy of around 0.65. SVM performed substantially worse than all the other algorithms, suggesting that hyperplane-based classifiers are less appropriate for CA based on keystroke mechanics. The results are relevant to researchers and practitioners aiming to design and implement effective and scalable CA systems.

We plan to analyze energy and resource usage as future work and compare them with the existing studies [24,36]. Other studies also present novel classifiers like artificial immune systems [37] or deep learning models [12], providing additional opportunities for comparison. Evidence on intrusion detection also showed that two-stage systems increase accuracy without compromising efficiency [38]. Keystroke mechanics can also be compared or complemented with other biometric data like facial recognition, fingerprint, or even novel ones like electrocardiogram-based authentication [39]. Similarly, intrasession features (e.g., user's clothes) can be considered as well as other behavioral data gathered from the mobile phone (e.g., apps running). Privacy is also a concern that can be mitigated with pseudonymization and anonymization approaches [40]. Finally, additional research into the number of keystroke events required to train accurate user models can also provide additional insights to researchers and practitioners. From a practical perspective, it may be necessary to deploy mobile CA systems avoiding the cold start problems inherent to ML solutions. There is also the possibility of users having different models under different conditions or devices, so intersession CA is also a promising research area.

Author Contributions: Conceptualization, L.d.-M. and J.-J.M.-H.; Data curation, J.J.-S. and C.C.; Formal analysis, J.J.-S. and C.C.; Funding acquisition, J.-J.M.-H.; Methodology, L.d.-M.; Project administration, L.d.-M.; Resources, C.P.-A.; Supervision, L.d.-M., J.-J.M.-H. and C.P.-A.; Validation, J.J.-S., C.C. and C.P.-A.; Writing—original draft preparation, L.d.-M., J.J.-S. and C.C.; Writing—review and editing, J.-J.M.-H. and C.P.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This project received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 826284.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the fact that all data used in this research are from a public external dataset. This study did not collect any new personal or sensitive information from any participant.

Informed Consent Statement: Participant consent was waived because all data comes from a public dataset. Description of participants and consent, if applicable, can be found in original the dataset. Please refer to the dataset source in Data Availability Statement below for more information.

Data Availability Statement: The data used in this research are from the HMOG dataset (http://www.cs.wm.edu/~qyang/hmog.html; accessed on 18 December 2019) licensed by The College of William and Mary for noncommercial, educational, and research purposes only. The College of William and Mary does not bear any responsibility for the analysis or interpretation of the HMOG dataset presented in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shukla, D.; Kumar, R.; Serwadda, A.; Phoha, V.V. Beware, Your hands reveal your secrets! In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 904–917.
2. Xu, Y.; Heinly, J.; White, A.M.; Monrose, F.; Frahm, J.-M. Seeing double: Reconstructing obscured typed input from repeated compromising reflections. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 1063–1074.
3. Aviv, A.J.; Gibson, K.; Mossop, E.; Blaze, M.; Smith, J.M. Smudge on smartphone touch screens. In Proceedings of the 4th USENIX Conference on Offensive Technologies, WOOT 10, Washington, DC, USA, 9 August 2010.
4. Menotti, D.; Chiachia, G.; Pinto, A.; Schwartz, W.R.; Pedrini, H.; Falcão, A.X.; Rocha, A. deep representations for iris, face, and fingerprint spoofing detection. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 864–879. [CrossRef]
5. Bonastre, J.-F.; Bimbot, F.; Boe, L.-J.; Magrin-Chagnolleau, I. Person authentication by voice: A need for caution. In Proceedings of the 8th European Conference on Speech Communication and Technology, EUROSPEECH 2003-INTERSPEECH 2003, Geneva, Switzerland, 1–4 September 2003.
6. Banerjee, S.; Woodard, D.L. Biometric Authentication and identification using Keystroke dynamics: A survey. *J. Pattern Recognit. Res.* **2012**, *7*, 116–139. [CrossRef]
7. Azenkot, S.; Zhai, S. Touch behavior with different postures on soft smartphone keyboards. In Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services, San Francisco, CA, USA, 21–24 September 2012; pp. 251–260.
8. Kim, K.-E.; Chang, W.; Cho, S.-J.; Shim, J.; Lee, H.; Park, J.; Lee, Y.; Kim, S. Hand grip pattern recognition for mobile user interfaces. In Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; Volume 2, pp. 1789–1794.
9. Ibrahim, A.; Thiruvady, D.; Schneider, J.-G.; Abdelrazek, M. The challenges of leveraging threat intelligence to stop data breaches. *Front. Comput. Sci.* **2020**, *2*. [CrossRef]
10. Smith-Creasey, M.; Rajarajan, M. A novel word-independent gesture-typing continuous authentication scheme for mobile devices. *Comput. Secur.* **2019**, *83*, 140–150. [CrossRef]
11. Clarke, N.L.; Furnell, S.M. Authenticating mobile phone users using keystroke analysis. *Int. J. Inf. Secur.* **2007**, *6*, 1–14. [CrossRef]
12. Volaka, H.C.; Alptekin, G.; Basar, O.E.; Isbilen, M.; Incel, O.D. Towards continuous authentication on mobile phones using deep learning Models. *Procedia Comput. Sci.* **2019**, *155*, 177–184. [CrossRef]
13. Serwadda, A.; Phoha, V.V.; Wang, Z. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In Proceedings of the IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), Arlington, VA, USA, 29 September–2 October 2013; pp. 1–8.
14. Siddiqi, M.A.; Pak, W. Optimizing filter-based feature selection method flow for intrusion detection system. *Electronics* **2020**, *9*, 2114. [CrossRef]
15. Bours, P.; Mondal, S. Continuous Authentication with Keystroke Dynamics. In *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*; Zhong, Y., Deng, Y., Eds.; Science Gate Publishing: Thrace, Greece, 2015; Volume 2, pp. 41–58.
16. Teh, P.S.; Zhang, N.; Teoh, A.B.J.; Chen, K. A survey on touch dynamics authentication in mobile devices. *Comput. Secur.* **2016**, *59*, 210–235. [CrossRef]
17. Shepherd, S.J. Continuous authentication by analysis of keyboard typing characteristics. In Proceedings of the European Convention on Security and Detection, Brighton, UK, 16–18 May 1995; pp. 111–114.
18. Ahmed, A.A.; Traore, I. Biometric recognition based on free-text keystroke dynamics. *IEEE Trans. Cybern.* **2014**, *44*, 458–472. [CrossRef]
19. Pisani, P.H.; Lorena, A.C. A systematic review on keystroke dynamics. *J. Braz. Comput. Soc.* **2013**, *19*, 573–587. [CrossRef]

20. Giuffrida, C.; Majdanik, K.; Conti, M.; Bos, H. I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Egham, UK, 10–11 July 2014; pp. 92–111.
21. Shuwandy, M.L.; Zaidan, B.B.; Zaidan, A.A.; Albahri, A.S. Sensor-Based mHealth authentication for real-time remote healthcare monitoring system: A multilayer systematic review. *J. Med. Syst.* **2019**, *43*, 33. [CrossRef] [PubMed]
22. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. RAMHU: A new robust lightweight scheme for mutual users authentication in healthcare applications. *Secur. Commun. Netw.* **2019**, *2019*, 3263902. [CrossRef]
23. Frank, M.; Biedert, R.; Ma, E.; Martinovic, I.; Song, D. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 136–148. [CrossRef]
24. Sitová, Z.; Šedĕnka, J.; Yang, Q.; Peng, G.; Zhou, G.; Gasti, P.; Balagani, K.S. HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 877–892. [CrossRef]
25. Yang, Q.; Peng, G.; Nguyen, D.T.; Qi, X.; Zhou, G.; Sitová, Z.; Gasti, P.; Balagani, K.S. A multimodal data set for evaluating continuous authentication performance in smartphones. In Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, Memphis, TN, USA, 3–6 November 2014; pp. 358–359.
26. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
27. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 5–32. [CrossRef]
28. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
29. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning. Data Mining, Inference and Prediction*; Springer: New York, NY, USA, 2009.
30. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
31. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
32. Junquera-Sánchez, J.; Cilleruelo-Rodríguez, C.; de-Marcos, L.; Martínez-Herráiz, J.J. JBCA: Designing an adaptative continuous authentication architecture. In *Advances in Physical Agents II*; Bergasa, L.M., Ocaña, M., Barea, R., López-Guillén, E., Revenga, P., Eds.; Springer International Publishing: Madrid, Spain, 2020; pp. 194–209.
33. Gascon, H.; Uellenbeck, S.; Wolf, C.; Rieck, K. Continuous authentication on mobile devices by analysis of typing motion behavior. In Proceedings of the Security 2014—Security, Protection and Reliability, Vienna, Austria, 19–21 March 2014; pp. 1–12.
34. Bell, R.M.; Koren, Y. Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl.* **2007**, *9*, 75–79. [CrossRef]
35. Al-Zewairi, M.; Almajali, S.; Ayyash, M. Unknown security attack detection using shallow and deep ANN classifiers. *Electronics* **2020**, *9*, 2006. [CrossRef]
36. Basar, O.E.; Alptekin, G.; Volaka, H.C.; Isbilen, M.; Incel, O.D. Resource usage analysis of a mobile banking application using sensor-and-touchscreen-based continuous authentication. *Procedia Comput. Sci.* **2019**, *155*, 185–192. [CrossRef]
37. Aljohani, N.; Shelton, J.; Roy, K. Continuous authentication on smartphones using an artificial immune system. In Proceedings of the 28th Modern Artificial Intelligence and Cognitive Science, Fort Wayne, IN, USA, 28–29 April 2017; pp. 171–174.
38. Reyes, A.A.; Vaca, F.D.; Castro Aguayo, G.A.; Niyaz, Q.; Devabhaktuni, V. A machine learning based two-stage wifi network intrusion detection system. *Electronics* **2020**, *9*, 1689. [CrossRef]
39. Zhang, Y.; Gravina, R.; Lu, H.; Villari, M.; Fortino, G. PEA: Parallel electrocardiogram-based authentication for smart healthcare systems. *J. Netw. Comput. Appl.* **2018**, *117*, 10–16. [CrossRef]
40. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. PAX: Using pseudonymization and anonymization to protect patients' identities and data in the healthcare system. *Int. J. Environ. Res. Public Health* **2019**, *16*, 1490. [CrossRef] [PubMed]