Outlier Detection for Risk-based User Authentication on Mobile Devices

Maria Papaioannou
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
m.papaioannou@greenwich.ac.uk

Georgios Zachos
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
g.zachos@av.it.pt

Georgios Mantas
Instituto de Telecomunicações
Aveiro, Portugal
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
gimantas@av.it.pt

Ismael Essop
Faculty of Engineering and Science, University of Greenwich
Chatham Maritime, UK
i.a.essop@gre.ac.uk

Firooz B Saghezchi
Instituto de Telecomunicações, Universidade de Aveiro
Aveiro, Portugal
firooz@av.it.pt

Jonathan Rodriguez
Faculty of Computing, Engineering and Science, University of South Wales
Pontypridd, UK
jonathan.rodriguez@southwales.ac.uk

***Abstract*—Mobile user authentication is the primary means of verifying the claimed identity of a user before granting access to resources on a mobile device. Common user authentication methods include passwords and biometrics. Despite the fact that passwords have been the most popular user authentication method for several decades, recent research suggests that they are no longer secure or convenient for mobile users due to several limitations that compromise both device security and usability. Biometric-based user authentication, on the other hand, is gaining popularity because it appears to strike a balance between security and usability. Such methods rely on human physical traits (physiological biometrics) or user involuntary actions (behavioral biometrics) for authentication. Risk-based user authentication using behavioral biometrics is particularly promising for mobile user authentication enhancing mobile authentication security while maintaining usability. In this context, we present an overview of mobile user authentication and discuss risk-based user authentication for mobile devices as a suitable approach to deal with the security vs. usability challenge. Afterwards, we test and evaluate a set of outlier detection algorithms for risk estimation in order to identify the most suitable ones for risk-based user authentication on mobile devices in terms of their accuracy and efficiency.**

Keywords—outlier detection, behavioral biometric-based user authentication, risk-based user authentication, mobile devices

I. INTRODUCTION

Mobile user authentication acts as the first line of defense verifying the identity of a mobile user and allowing access to resources on a mobile device [1]–[3]. According to NIST [4], the most common user authentication mechanisms are password-based and biometric-based. Nevertheless, traditional password-based user authentication methods are no longer considered secure or practical for mobile users [5]–[8]. These methods authenticate anyone who possesses the correct credentials, regardless of whether they are the legitimate users or not [9]–[11]. Additionally, mobile users often struggle to remember complex passwords, resulting in weak passwords that are easily guessed or stolen through various attacks such as shoulder-surfing, dictionary or guessing attacks [12].

On the other hand, biometric-based user authentication is being increasingly recognized as a more secure and usable method compared to traditional password-based authentication [6], [13]. Biometrics can be divided into two categories: physiological and behavioral. Biometric-based user authentication relying on physiological biometrics utilizes mobile user´s human physical characteristics such as fingerprints or facial traits to authenticate them, while biometric-based user authentication relying on behavioral biometrics makes use of mobile user´s involuntary actions such as gait or typing patterns to verify their claimed identity [7]. Mobile device manufacturers like Samsung, Apple, and Nokia have already started embedding sensors in their devices to allow for physiological biometric authentication. However, although physiological biometrics are considered secure since they are unique, they have been proven to be vulnerable to security attacks including impersonation. More specifically, researchers have shown that physiological biometric-based user authentication schemes can be hacked easily with not very sophisticated algorithms and a cheap equipment using photos of the legitimate user extracted from social media [14], [15]. Behavioral biometrics, on the other hand, are considered more secure and accurate as they are unique and cannot be copied, lost or stolen [7]. On top of that, they can be gathered easily and cost-effectively through in-built sensors in mobile devices [16], [17] and can be deployed as an additional layer of authentication, establishing multifactor authentication without affecting the usability of the device [1]–[3], [8], [18].

In particular, risk-based user authentication relying on behavioral biometrics has emerged as a potential solution to enhance mobile authentication security without sacrificing usability [19]. This type of authentication mechanism verifies the identity of a mobile user in real-time based on a risk score without interrupting their usual activity [7]. In our previous works [11], [20]–[22], we: (i) provided a thorough related work on mobile user authentication, (ii) introduced the security vs. usability challenge, and (iii) presented the concept of the risk-based user authentication for mobile devices. On top of that, we presented the design of a risk-based adaptive user authentication mechanism that balances security and usability in mobile user authentication, ensuring

continuous user authentication behind-the-scenes and invisible to the user. In addition, we trained and tested popular classification algorithms for risk-based authentication, namely Decision Trees (DT), k-Nearest Neighbour (k-NN), Naïve Bayes (NB) and Support Vector Machine (SVM), over the ‘‘HuMIdb’’ dataset [23], [24] using ten-fold cross validation to identify the most appropriate ones for the proposed mechanism. Nevertheless, the evaluation results showed impact of overfitting. Therefore, the current paper aims to investigate the concept of outlier detection for risk-based user authentication on mobile devices to overcome this challenge of overfitting. The goal is to test outlier detection algorithms found in the literature and evaluate them to identify the most appropriate ones that can also be applied to the proposed mechanism in [22].

Following the Introduction, the rest of the paper is structured as follows. Section II provides an outline of risk-based user authentication on mobile devices. In Section III, the prevalent outlier detection algorithms for behavioral biometric-based user authentication are discussed, and in Section IV, the performance evaluation of these outlier detection algorithms is presented. Finally, Section V concludes the paper.

II. Risk-based User Authentication on Mobile Devices

Risk-based user authentication methods enable a mobile device to verify the legitimacy of a user without the need for the user to explicitly provide authentication [25]. In [7], the authors define risk-based user authentication as a "continuous evaluation of whether to accept or reject a user's authentication based on their behavior and the risk associated with their actions." This determination is made by comparing the real-time risk score to the stored scores in the user's risk profile. If necessary, the system prompts the user for re-authentication, accordingly, as illustrated in Fig. 1.

Undoubtedly, the risk estimation component plays a crucial role in risk-based user authentication (RBA) mechanisms, as it is responsible for calculating an accurate real-time risk score based on the available information from the user's contextual information and behavior, as well as the device's contextual information. Wiefling et *al.* emphasize that the accurate estimation of a risk score has a significant impact on both the usability and security of RBA [26]. Commonly used methods for estimating the risk score include qualitative risk assessments (RA) [27], and the most widely used mathematical formula to represent it is the following:

$$Risk\ Score = Likelihood \times Impact.$$

Then, the estimated risk scores are transformed into a "human readable" format (i.e., high, medium, or low risk). Nevertheless, in these qualitative approaches, the risk scores are always rated subjectively, and this makes them unsuitable for real-world cybersecurity solutions [28]. Therefore, there is the tendency to move in the direction of more quantitative risk estimation approaches. In the literature, various machine

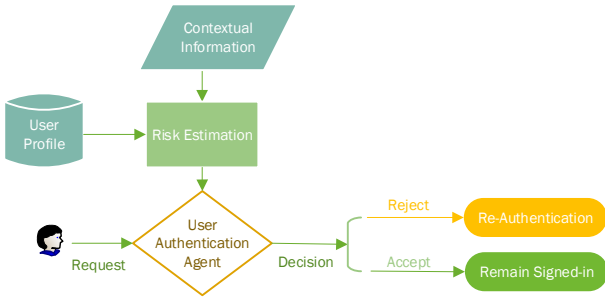

Fig. 1. An Overview of Risk-based User Authentication.

learning classification algorithms such as Naïve Bayes [29], and decision trees [30], have been proposed for quantitative risk estimation for risk-based user authentication. In our most recent work [22], we trained and tested the k-Nearest Neighbour (k-NN), Decision Trees (DT), Support Vector Machine (SVM), and Naïve Bayes (NB), which constitute the most popular classification algorithms for risk-based authentication. During the training process, the evaluation results demonstrated that these models became closely related with particular features and thus, they showed impact of overfitting. To overcome the challenge of overfitting, we considered the concept of outlier detection. In [26], the authors used outlier detection algorithms including the isolation forest and minimum covariance determinant, for risk-based authentication on a real-world large-scale online service. Therefore, in the current paper, our aim is to test a set of four popular outlier detection algorithms to identify the most efficient ones for risk-based user authentication on mobile devices.

III. Outlier Detection Algorithms for Risk-Based User Authentication

Outlier detection machine learning algorithms involve training data that are contaminated with a small proportion of outliers (i.e., observations that differ significantly from the normal class). These outliers may be associated with malicious users or low-quality data, such as samples with missing entries. Then, outlier detection estimators attempt to fit the regions with the most densely populated training data, overlooking the deviant observations. Various applications including user authentication for mobile devices, require the ability to determine whether a new observation is an inlier, meaning that it belongs to the same distribution as the existing observations, or an outlier, indicating that it differs from the existing observations [31].

In the context of risk-based user authentication relying on behavioral biometrics, the typical scenario involves single-user mobile devices, where it is necessary to differentiate between a known legitimate user and an unknown malicious user. Toward this direction and according to the literature [26], outlier detection algorithms, also known as one-class classifiers, have attracted the attention of researchers and demonstrated significant benefits for user authentication based on behavioral biometrics.

In fact, the main advantage of one-class outlier detection algorithms is that they require only a minimal number of samples from the impostor's class. For instance, in our case, we specifically chose a 9:1 ratio. With the limited availability of data for behavioral biometrics, coupled with the rapid evolution of data acquisition quality for mobile computing devices, unsupervised anomaly detection methods, such as outlier detection, are a suitable modelling strategy for risk-based user authentication based on behavioral biometrics. In contrast, supervised models are difficult to utilize in a real-world user authentication application, as there are not enough negatively labelled samples per-user. In the remainder of this section, we provide a description of the four outlier detection algorithms we employed for risk-based user authentication on mobile devices, namely Isolation Forest, Minimum Covariance Determinant, AutoEncoder, and Outlier Detection with KNN.

A. *Isolation Forest*

Isolation Forest (IF) is an outlier detection algorithm that distinguishes outliers from the normal data by determining the distance of a new data point to the rest of the data points, rather than modeling the normal points like other common machine learning algorithms [32], [33]. IF uses an ensemble of decision trees to accomplish this. The algorithm's core principle is that outliers in a dataset are typically easier to separate (isolate) from the rest of the data samples than normal points. To isolate a data point, IF recursively creates partitions (i.e., lines orthogonal to the origin) on the sample by randomly selecting an attribute and then randomly selecting a split value for the attribute from the allowed minimum and maximum values. The algorithm assigns higher outlier scores to data points that required fewer splits to be isolated. An illustration of how IF isolates a normal point and an outlier is given in Fig. 2.

B. *Minimum Covariance Determinant*

Minimum Covariance Determinant (MCD) is an outlier detection algorithm that estimates the mean and covariance matrix of a set of data in such a way as to minimize the impact of outliers [34]. The main idea is to calculate these parameters (i.e., the mean and covariance matrix) from a subset of the whole data that has been chosen to be free of anomalies. For this, the MCD algorithm starts by taking a set of subsamples of data of a given size and determining the mean and covariance matrix for each subsample. Afterwards, the algorithm stores the estimates for the subset whose covariance matrix appears to have the smallest determinant. The purpose of minimizing the determinant is that essentially the determinant of the covariance matrix measures how wide the distribution is [34]. Thus, the MCD algorithm chooses the smallest determinant (i.e., the most densely distributed data subsample). In this way, the MCD algorithm excludes the outliers that are likely to be found further from the rest of the data. An illustration of MCD algorithm is given in Fig. 3.

C. *AutoEncoder*

AutoEncoder comprises an unsupervised artificial neural network and functions by compressing the original data samples into a shortcode without taking into consideration any noise [35]. Afterwards, the algorithm uncompresses that shortcode to generate an output as close as possible to the original data. Although this algorithm is mainly used for image classification, it can be also effectively used for outlier detection [35]. In this case, if the model is trained with a given dataset, outliers will give a higher reconstruction error, so it will be easier to be detected by AutoEncoder.

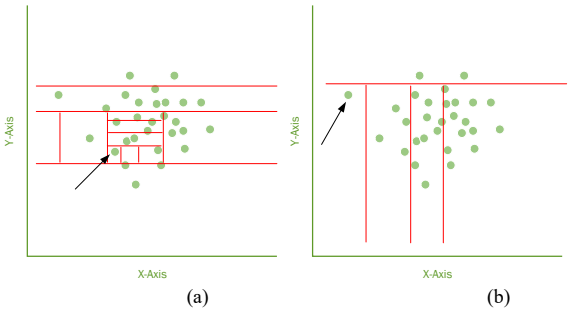

Fig. 2. An Illustration of Isolating: (a) a Normal Point; and (b) an Outlier using Isolation Forest Outlier Detection Algorithm.

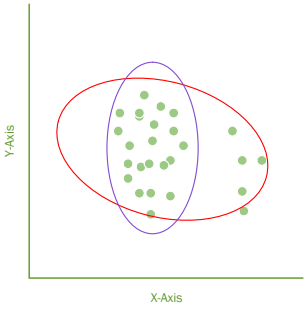

Fig. 3. An Illustration of Minimum Covariance Determinant Algorithm.

D. *Outlier Detection with KNN*

Although KNN is a popular supervised classification algorithm, it can be also efficiently used as unsupervised outlier detection algorithm. In this case, the distance to the kth nearest neighbor is considered as a local density estimate, also known as the outlier score in outlier detection. As such, the larger the distance to the kth nearest neighbor is, the local density is lower, and thus it is more likely the new point to be an outlier. Although this algorithm is quite simple, it has proved to work very effectively outperforming more recent and more complex approaches, according to a large scale experimental analysis [36].

IV. Performance Evaluation of Outlier Detection Algorithms for Risk-Based User Authentication

We used ten-fold cross validation to train and test the above-mentioned outlier detection algorithms (i.e., isolation forest, minimum covariance determinant, autoencoder and outlier detection with KNN) over the generated dataset derived from the HuMIdb dataset [23], [24]. In particular, the generated dataset includes all the records of the first user (i.e., user000) and some records of the second user (i.e., user001) of the HuMIdb dataset. Furthermore, in our experiments, we considered the following: i) the first user

(i.e., user000) is benign, and ii) the second user (i.e., user001) is malicious. Moreover, the ratio between the records of the first user (i.e., user000) and the second user (i.e., user001) dataset was 9:1 in favor of the records of the first user (i.e., user000). The reason for this ratio in the records stems from the requirement of the outlier detection algorithms where the number of the records belonging to the malicious class must be smaller than the number of the records belonging to the benign class. We chose specifically the 9:1 ratio because the implementations of the outlier detection algorithms in the employed PyOD [37] python library utilize this exact default value (i.e., 0.1) for a related parameter (i.e., "contamination" parameter equal to 0.1). In addition, we modified the generated dataset by deleting all features which were related to humidity, light, proximity, temperature, gps, wifi, Bluetooth, and microphone in the "HuMldb" dataset files because they: (a) suffered from empty entries, (b) involved alphanumeric values where further processing was not possible, and/or (c) were depending on specific device characteristics (e.g., MAC address) that already had fixed values. For the remainder of this section, this part of the dataset will be referred to as generated dataset. The performance of the outlier detection algorithms was evaluated on the generated dataset based on the evaluation metrics of accuracy, recall, precision, and F1-score. The Minimum Covariance Determinant demonstrated a significantly high performance among the four outlier detection algorithms.

A. Dataset pre-processing and normalization

In principle, before training and testing outlier detection algorithms with the available datasets, they are required to be properly prepared. In particular, data preparation comprises two specific steps: (i) data pre-processing; and (ii) data normalization. The data pre-processing step involves removing the unnecessary features and converting the nominal values of the categorical features to numeric values. However, in our case, the values of all features were already numeric and thus, no redundant features needed to be removed. Therefore, we omitted the data pre-processing step regarding the generated dataset. Next, the data normalization step was performed to the numeric values of each feature.

Generally, the outlier detection algorithms may demonstrate inaccurate results if the values of a feature are considerably larger/smaller in comparison to the values of other features. Therefore, the data normalization step is essential so that it is ensured that features with significantly larger values will not outweigh features with small values. This is achieved by performing a min-max normalization process on every feature to guarantee that the values of all features are placed within the range of [0.0, 1.0]. The following equation describes the normalization process:

$$z = (x - xmin) / (xmax - xmin)$$

where x is the value before scaling, xmax and xmin are the maximum and minimum values of the feature, and z is the normalized value (i.e., after scaling) respectively.

B. Training process of outlier detection algorithms

We trained and tested the outlier detection algorithms over the generated dataset. Initially, we divided the dataset into two parts: (a) the train part consisting of 80% of the dataset and (b) the test part consisting of 20% of the dataset. The train part was employed for the training and evaluation of the outlier detection algorithms, while, on the other hand, the test part was held back for further evaluation of the models with unseen data. According to [38], the percentage split of 80% train data-20% test data was decided as the best ratio so that the overfitting problem can be avoided. Then, using the ten-fold cross validation method, we performed the training process of each outlier detection algorithm. According to the ten-fold cross validation method, we divided the training dataset into ten subsets of equal size and randomly chose the records of each subset. Afterwards, the training process was repeated ten times. Each time, nine of the ten subsets were used to train the outlier detection algorithms and the remaining subset was employed for validation.

In our tests, we used the Python language version 3.9.7 was used, as well as the PyOD [37] library and the Scikit-Learn [39] library. We used certain functions of the PyOD library and the Scikit-Learn library, and developed an appropriate Python script, utilizing these functions so that we can perform the training and testing of the four selected outlier detection algorithms.

Table I. Summary of the hyperparameters of each outlier detection algorithm.

Algorithm	Hyperparameters
IF	The "contamination" parameter was set to 0.1.
MCD	The "contamination" parameter was set to 0.1.
AutoEncoder	1) The "contamination" parameter was set to 0.1. 2) The "hidden_neurons" parameter was set to the value [8, 4, 4, 8].
Avg_KNN	1) The "contamination" parameter was set to 0.1. 2) The "n_neighbors" parameter was set to 5. 3) The "method" parameter was set to "mean".

Table II. Evaluation metrics for outlier detection for the "HuMIdb" dataset.

Algorithm	Accuracy	Precision	Recall	F1-Score
IF	0.93	1.00	0.93	0.96
MCD	0.97	1.00	0.97	0.98
AutoEncoder	0.91	1.00	0.91	0.95
Avg_KNN	0.91	1.00	0.92	0.96

C. Performance evaluation results

We averaged the results of the ten folds and produced the performance results of the outlier detection algorithms [38]. Table I provides details regarding the hyperparameters of each outlier detection algorithm. In Fig. 4 and Table II, the numerical results of the evaluation metrics for the selected outlier detection algorithms are shown. Among the four outlier detection algorithms, we can easily observe that MCD demonstrates a significantly high performance in accuracy (i.e., 0,97). As far as the rest of the evaluation metrics (i.e.,

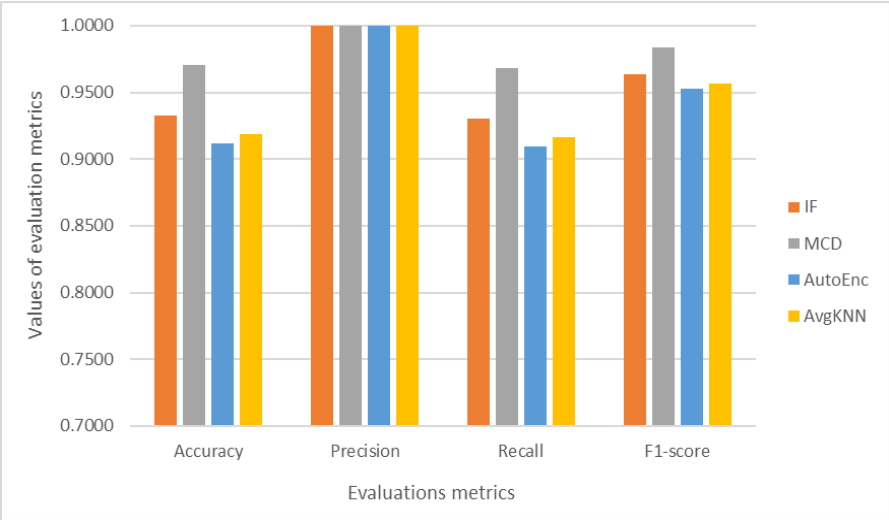

Fig. 4. Evaluation metrics for outlier detection for the ``HuMIdb" dataset.

precision, recall, and F1-score) are concerned, the "MCD" outlier detection algorithm continues to demonstrate the best performance in both recall and F1-score (i.e., 0,97 and 0,98 respectively). Moreover, it is worth mentioning that all four outlier detection algorithms exhibit a perfect score for the precision evaluation metric.

V. Conclusions

For several decades, password-based approaches have been the most common method used for user authentication on mobile devices. However, recent studies suggest that passwords are no longer considered a secure or convenient method for mobile users due to various limitations. Therefore, there is a growing need to develop and implement more secure and user-friendly methods for user authentication. One promising approach is user authentication based on "something the user is," based on physiological and/or behavioral biometrics. In particular, risk-based user authentication using behavioral biometrics is gaining attention as a potential solution to improve mobile authentication security while without sacrificing usability. In the current paper, our aim was to investigate the concept of outlier detection for risk-based user authentication on mobile devices and thus, we focused on testing outlier detection algorithms, found in the literature, and evaluating them. In particular, we trained and tested four outlier detection algorithms (i.e., isolation forest, minimum covariance determinant, autoencoder and outlier detection with KNN) over a dataset that we generated based on the well-known "HuMIdb" dataset in order to identify the most appropriate ones for risk-based user authentication on mobile devices. It is worthwhile to highlight that among the four outlier detection algorithms, the MCD algorithm demonstrated the best performance in terms of accuracy, recall and F1-score. As our next step, we plan to continue the training and testing of more outlier detection algorithms over the same training part, using 10-fold cross validation as well as various combinations of hyperparameters for each outlier detection algorithm so that the best hyperparameters for each algorithm can be determined. After we have identified the best hyperparameters for each outlier detection algorithm, one final performance evaluation will be performed over the same testing part so that we can acquire more realistic performance metrics over unseen data.

References

[1] G. Beier, P. Hoffman, and S. Shorter, "Information System Security Best Practices for UOCAVA- Supporting Systems (NISTIR 7682)," *Natl. Inst. Stand. Technol. (NIST)*, 2011.

[2] M. Papaioannou *et al.*, "A survey on security threats and countermeasures in Internet of Medical Things (IoMT)," *Trans. Emerg. Telecommun. Technol.*, no. May, pp. 1–15, 2020.

[3] M. Papaioannou, J. C. Ribeiro, V. Monteiro, V. Sucasas, G. Mantas, and J. Rodriguez, "A privacy-preserving user authentication mechanism for smart city mobile apps," in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (IEEE CAMAD)*, 2021, pp. 1–5.

[4] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "NIST 800-63-3: Digital Identity Guidelines," *NIST Spec. Publ.*, p. 68, 2017.

[5] M. Harbach, E. von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith, "It's a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception," *SOUPS '14 Proc. Tenth Symp. Usable Priv. Secur.*, pp. 213–230, 2016.

[6] I. Stylios, S. Kokolakis, O. Thanou, and S. Chatzis, "Behavioral biometrics & continuous user authentication on mobile devices: A survey," *Inf. Fusion*, vol. 66, no. February 2020, pp. 76–99, 2021.

[7] S. Gupta, A. Buriro, and B. Crispo, "Demystifying authentication concepts in smartphones: Ways and types to secure access," *Hindawi Mob. Inf. Syst.*, vol. 2018, 2018.

[8] F. Pelekoudas-Oikonomou *et al.*, "Blockchain-Based Security Mechanisms for IoMT Edge Networks in IoMT-Based Healthcare Monitoring Systems," *Sensors*, vol. 22, no. 7, 2022.

[9] M. Papaioannou, G. Zachos, G. Mantas, and J. Rodriguez, "Novelty Detection for Risk-based User Authentication on Mobile Devices," in *IEEE Global Communications Conference*, 2022, p. Accepted to be

published.

[10] M. Papaioannou, F. Pelekoudas-oikonomou, G. Mantas, E. Serrelis, J. Rodriguez, and M. Fengou, “A Survey on Quantitative Risk Estimation Approaches for Secure and Usable User Authentication on Smartphones,” *Sensors 2023*, vol. 23, 2979, 2023.

[11] M. Papaioannou, G. Mantas, D. Lymberopoulos, and J. Rodriguez, “User authentication and authorization for next generation mobile passenger ID devices for land and sea border control,” in *12th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2020*, 2020, pp. 8–13.

[12] J. Zhang, X. Luo, S. Akkaladevi, and J. Ziegelmayer, “Improving multiple-password recall: An empirical study,” *Eur. J. Inf. Syst.*, vol. 18, no. 2, pp. 165–176, 2009.

[13] NIST, “Biometrics - Ensuring Successful Biometric Systems,” 2008.

[14] J. Titcomb, “Hackers claim to beat iPhone X’s face id in one week with 115 mask,” 2017. [Online]. Available: http://www.telegraph.co.uk/technology/2017/11/13/hackers-beat-iphone-xs-face-oneweek-115-mask/. [Accessed: 07-Jan-2023].

[15] A. Charles, “The guardian-iPhone 5S fingerprint sensor hacked by Germany’s Chaos Computer Club,” 2013. [Online]. Available: https://www.theguardian.com/technology/2013/sep/22/apple-iphonefingerprint-scanner-hacked. [Accessed: 10-Jan-2023].

[16] S. Gupta, R. Kumar, M. Kacimi, and B. Crispo, “IDeAuth: A novel behavioral biometric-based implicit deauthentication scheme for smartphones,” *Pattern Recognit. Lett.*, vol. 157, no. March, pp. 8–15, 2022.

[17] T. Zhu *et al.*, “RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild,” *IEEE Trans. Mob. Comput.*, vol. 19, no. 2, pp. 466–483, 2020.

[18] J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez, and R. A. Abd-Alhameed, “HIDROID: Prototyping a behavioral host-based intrusion detection and prevention system for android,” *IEEE Access*, vol. 8, pp. 23154 – 23168, 2020.

[19] S. Wiefling, M. Dürmuth, and L. Lo Iacono, “Verify It’s You: How Users Perceive Risk-Based Authentication,” *IEEE Secur. Priv.*, vol. 19, n, no. December, pp. 47–57, 2021.

[20] M. Papaioannou, G. Mantas, A. Essop, P. Cox, I. E. Otung, and J. Rodriguez, “Risk-based adaptive user authentication for mobile passenger ID devices for land/sea border control,” in *IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2021, pp. 1–6.

[21] M. Papaioannou, G. Mantas, and J. Rodriguez, “Risk-based user authentication for mobile passenger ID devices for land and sea border control,” in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 180–185.

[22] M. Papaioannou, G. Zachos, I. Essop, G. Mantas, and J. Rodriguez, “Towards a Secure and Usable User Authentication for Mobile Passenger ID Devices for Land/Sea Border Control,” *IEEE Access*, vol. 10, pp. 38832–38849, 2022.

[23] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar, “BeCAPTCHA: Bot detection in smartphone interaction using touchscreen biometrics and mobile sensors,” *arXiv Prepr. arXiv2005.13655*, no. May, 2020.

[24] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and I. Bartolome, “BeCAPTCHA: Detecting human behavior in smartphone interaction using multiple inbuilt sensors,” *arXiv Prepr. arXiv2002.00918*, 2020.

[25] N. Clarke, “Frictionless user authentication,” in *Encyclopedia of Cryptography, Security and Privacy*, S. Jajodia, P. Samarati, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 1–5.

[26] S. Wiefling, P. R. Jørgensen, S. Thunem, and L. Lo Iacono, “Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service,” *ACM Trans. Priv. Secur.*, vol. 1, no. 1, pp. 1–35, 2022.

[27] T. Lederm and N. L. Clarke, “Risk assessment for mobile devices,” in *International Conference on Trust, Privacy and Security in Digital Business*, 2011, pp. 210–221.

[28] D. W. Hubbard and R. Seiersen, *How to Measure Anything in Cybersecurity Risk*. Wiley, 2016.

[29] M. Heydari, A. Mylonas, V. Katos, E. Balaguer-Ballester, V. H. F. Tafreshi, and E. Benkhelifa, “Uncertainty-aware authentication model for fog computing in IoT,” in *Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019, pp. 52–59.

[30] J. Spooren, D. Preuveneers, and W. Joosen, “Mobile device fingerprinting considered harmful for risk-based authentication,” in *Proceedings of the Eighth European Workshop on System Security*, 2015, pp. 1–6.

[31] scikit-learn developers, “Novelty and Outlier Detection.” [Online]. Available: https://scikit-learn.org/stable/modules/outlier_detection.html.

[32] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation-based anomaly detection,” *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, 2012.

[33] R. Kumar, P. P. Kundu, and V. V. Phoha, “Continuous authentication using one-class classifiers and their fusion,” *2018 IEEE 4th Int. Conf. Identity, Secur. Behav. Anal. ISBA 2018*, vol. 2018-Janua, pp. 1–8, 2018.

[34] J. Hardin and D. M. Rocke, “Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator,” *Comput. Stat. Data Anal.*, vol. 44, no. 4, pp. 625–638, 2004.

[35] S. Zhu *et al.*, *Outlier Analysis Second Edition*, vol. 24, no. 2. 2017.

[36] G. O. Campos *et al.*, *On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study*, vol. 30, no. 4. Springer US, 2016.

[37] Y. Zhao, Z. Nasrullah, and Z. Li, “PyOD: A python toolbox for scalable outlier detection,” *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, 2019.

[38] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O’Reilly Media, 2019.

[39] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.