Spoofing Detectability as a Property of Biometric Characteristics

Tommaso Zoppi[1], Enrico Schiavone[2], Irene Bicchierai[2],
Francesco Brancati[2], Andrea Bondavalli[1]

[1] *Department of Mathematics and Informatics, University of Florence, Viale Morgagni 65 50142 Florence, Italy*
[2] *Resiltech s.r.l., Piazza Nilde Iotti, 25 - 56025 Pontedera (Pisa), Italy*

Abstract

Regardless of the application domain, adversaries may conduct spoofing attacks in order to bypass an authentication system. The difficulty of fooling a biometric sensor, known as circumvention; can be paired with an additional property based on the easiness of identifying ongoing presentation attacks which could help selecting the most suitable characteristic(s) when designing a biometric system. To such extent, this paper proposes spoofing detectability, as a property of biometric characteristics, to indicate the likelihood of detecting ongoing presentation attacks aiming at overcoming authentication mechanisms. We define and then quantitatively estimate spoofing detectability through unsupervised anomaly detection on publicly available biometric datasets, collecting metric scores which are then converted into the Low, Medium, High categories for 8 different biometric characteristics. We built our results upon unsupervised algorithms as they represent the most suitable answer to the detection of zero-day attacks. Alongside with our experimental process, we show the intrinsic relevance of spoofing detectability to complement circumvention. As a final contribution of the paper, we show how to embed an anomaly-based spoofing detection module into an authentication system for runtime support.

Keywords

Sensor Spoofing, Biometrics, Anomaly Detection, Presentation Attack, Security, Intrusion Detection

1. Introduction

In many critical systems and applications, only authorized users should interact with a given system. User authentication, which is the process of *verifying the identity claimed by or for a human entity* [39], is designed for this purpose. Traditional authentication approaches usually rely either on something the user ***knows*** (knowledge-based, e.g., passwords or PINs), or something the user ***has*** (e.g., security token). Instead, in the last two decades, research moved onto authentication mechanisms based on biometric characteristics, or rather what each user ***is*** or ***does***. The use of biometrics and its implications has been widely explored and expanded in literature [29], [33], [36], [39], and many different biometric characteristics have been proposed.

The adequate biometric characteristic for a given system should be carefully selected according to specific criteria. These criteria usually derive from intrinsic properties of characteristics [29], [36], namely: universality, distinctiveness, permanence, collectability, performance, acceptability, and circumvention. Nevertheless, in some cases it is possible, (and often recommended), to select more

Proceedings Name, Month XX–XX, YYYY, City, Country
EMAIL: tommaso.zoppi@unifi.it (A.1); enrico.schiavone@resiltech.com (A.2), irene.bicchierai@resiltech.com (A.3), francesco.brancati@resiltech.com (A.4), bondavalli@unifi.itemail2@mail.com (A.5)
ORCID: 0000-0001-9820-6047 (A.1);


than a single biometric characteristic, originating a multi-modal biometric authentication system [37], [38], [35], [40].

Clearly, authentication systems based on biometrics may still make mistakes, either authenticating impostors, or preventing legitimate users to interact with the system. In particular, there is a wide range of studies [29], [30], [34], [36], [38] that devise possible threats as presentation or spoofing attacks. As described in [42], a presentation or sensor spoofing attack is *an attempt to circumvent a biometric system by forging the trait of an authorized person and presenting it to the sensor*. Most biometric characteristics can be forged with an adequate effort, even if they are hard to circumvent [31], [32], [41]. This demands for spoofing detection mechanisms tailored according to the biometric characteristic(s) selected for the system.

Depending on the specific biometric characteristic, detecting presentation attacks or, more in general, threats to the biometric sample collection phase, can be very difficult. Therefore, this paper introduces spoofing detectability, an additional property of a biometric characteristic to indicate the easiness of identifying an ongoing presentation attack that overcame the comparison module and the available defenses, regardless of the circumvention value that was assigned to that characteristic. We assume that malicious activities leave some traces in the features extracted from the biometric samples. Observing these alterations contribute to detect spoofing attacks and, consequently, estimate spoofing detectability.

More in detail, we conduct a quantitative estimation of spoofing detectability through anomaly detection [9], [10], [55], recognized as the most suitable answer to detect unknown faults or *zero-day* attacks to a biometric authentication system. This way, we are estimating the detectability of presentation attacks without assuming any previous knowledge about them. We select different unsupervised anomaly detection algorithms, which we then apply to public datasets comprising feature values of the following biometric characteristics: face, fingerprint, voice, keystroke, heart rate variability, electrodermal activity, human gait, and hand gesture. We collect, analyze, and discuss the results of our experimental campaign, elaborating on the synergy of spoofing detectability and circumvention properties. Ultimately, we show how spoofing detectability can provide runtime support to traditional systems with a module that runs independently from the comparison module, helping to decide on authentication.

The paper develops as follows: Section 2 describes anomaly detection and presents families of unsupervised algorithms before introducing biometric systems. Spoofing Detectability is motivated and defined in Section 3, while Section 4 expands on our experimental campaign. Section 5 presents and discusses experimental results, used then in Section 6 to derive spoofing detectability. The role of spoofing detectability at runtime is debated in Section 7, while Section 8 closes the paper.

2. Anomaly Detection and Biometrics

2.1. Anomaly Detection

In the paper we refer to data point as the values of the features extracted from the biometric samples that the user provides to the sensor. Each data point is composed of f feature values, which are processed to determine whether the data point exhibits anomalies. More in detail, anomalies are *rare data points* showing *patterns that do not conform to a well-defined notion of normal behavior* [9]. Consequently, anomaly detection algorithms target the correct and complete definition of a normal behavior, and then identify anomalies by difference. Note that this detection mechanism – similarly to others – assumes that errors or attacks manifest as observable deviations with respect to the expected behavior [10], [51]. If an event happens without observable behavior e.g., perfectly obfuscated attack, anomaly-based detectors will not be able to successfully operate.

Different anomaly detectors may be instantiated depending on the nature of the target system [9], [55] and monitored data. If labeled training data is available, (semi-)supervised anomaly detection algorithms may be adopted [11]. Otherwise, the only option is to use unsupervised anomaly detection [10], which noticeably allows dealing with unknown, zero-day attacks. The potential to detect previously unknown or unseen attacks i.e., zero day attacks, is a critical asset for anomaly-based

detectors, as it permits covering weaknesses of signature-based mechanisms. Consequently, in the remainder of the paper we consider only unsupervised algorithms.

Throughout years, unsupervised algorithms have been studied and compared to derive similarities or differences. They have been grouped by related studies [9], [10], [8] into six main families, namely clustering [17], statistical [12], classification [15], neighbor-based [14], density-based [16], and angle-based [13]. Algorithms belonging to each family have their own peculiar aspects; however, it is worth noticing that there are some unavoidable semantic overlaps among families. A nearest-neighbour search may be embedded into other algorithms e.g., the angle-based FastABOD [13], while density measures may be built on top of clustering procedures i.e., LDCOF [43].

2.2. Biometric Authentication Systems

Traditional authentication mechanisms are either *knowledge-based*, or *possession-based.* Instead, a biometric system [36] is "*a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database*". Biometric systems are used in verification or identification modes and applied to multiple contexts, as i.e., forensics, authentication, access control.

Biometric characteristics are divided into: i) physiological, which are related to the shape of the body e.g., fingerprint, palm veins, face, DNA, palmprint, hand geometry, iris, and ii) behavioral, related to the behavior of a person, e.g., keystroke, or gait. Each of these characteristics is described through properties [29], [36] as follows.

- *Universality*: each person should have the characteristic.
- *Distinctiveness*: any two persons should be sufficiently different in terms of the characteristic.
- *Permanence*: the characteristic should be sufficiently invariant over a period of time.
- *Collectability*: the characteristic can be measured quantitatively
- *Performance*, the achievable recognition accuracy and speed, required resources, as well as the operational and environmental factors that affect them;
- *Acceptability*, the extent to which people are willing to use the biometric characteristic in their daily lives;
- *Circumvention*, which reflects how easily the system can be fooled using fraudulent methods.

These properties drive the selection of the most appropriate characteristic for a given system. We remark that collecting biometric data may raise privacy concerns; recently the GDPR [50] explicitly stated that biometrics are personal information and therefore must be protected "by design" and "by default".

As summarized in [40], a biometric system is called multimodal [36] or multi-biometric if it relies on: i) multiple different characteristics, i.e. face and iris, or ii) multiple acquisitions of the same characteristic, e.g., fingerprint-based systems where the user provides several fingers to the sensor. These systems have various advantages [35]; namely they: i) guarantee better accuracy in recognition, ii) provide redundancy, iii) force attackers to forge multiple characteristics simultaneously. On the other hand, multimodality may reduce usability, and increase computational cost, and resources needed to fulfill the authentication process as well as its duration.

2.3. Related Works on Sensor Spoofing and Anomaly Detection

Regardless of the multi-modality of a biometric authentication system, attackers may want to impersonate authorized users to gain access to a system [61]. Most of those presentation attacks are known as sensor spoofing, where the attacker forges biometric samples to fool the authentication system. Those attacks may compromise confidential information in many applications such as video surveillance [56], biometric identification [57], face indexing in social media [58], access to smartphones [59], iris recognition [31], physical access control through fingerprints [41], or even recognition of passengers in airports [60].

As a consequence, sensor spoofing started to be investigated and precisely characterized by surveys. The encyclopedia of biometrics [29] provides a comprehensive reference to concepts, technologies, issues, and trends in the field of biometrics. Particularly, it defines sensor spoofing as a method of attacking biometric systems where an artificial object is presented to the biometric sample acquisition system that imitates the biological properties the system is designed to measure, so that the system will not be able to distinguish the artifact from the real biological target. Different attacks are surveyed in the paper, which also lists common countermeasures as analysis of the resolution of biometric data, measurement of variation in the biometric property over short time durations, simultaneous measurement of a second biometric property (multi-biometrics), and many others. Despite being 20 years old, the paper [62] still provides another nice overview on the vulnerability of attacks at the sensor level, including the spoof attack or use of an artificial biometric sample to gain unauthorized access. In [30], authors seek to present a broader and more practical view of biometric system attack vectors, placing them in the context of a risk-based systems approach to security and defenses. Similarly, the work [42] traces attack trees for different spoofing attacks to derive attack paths specific for each malicious activity.

In the last decade, researchers, practitioners and industries started adopting Machine Learning (ML) algorithms as spoofing detectors. Distance-based methods were proven to be effective in analyzing features extracted from biometric samples for detection purposes [34], while One-Class Support Vector Machines were used in [63] as spoofing detectors. Moreover, deep learners proven to be a suitable answer to detect spoofing attacks either by learning more accurate representations of the biometric sample [57] or by implementing a complete detector [58].

3. Spoofing Detectability

In order to protect the system against presentation / sensor spoofing attacks originated by the forgery of biometric characteristics [42], choosing the characteristics that accounts for low circumvention may not be sufficient.

Conducting a presentation attack by forging a fingerprint e.g., fabricating artificial, "gummy", fingerprints [41], or by simulating the voice of an authorized person, can be relatively easy (high circumvention in [36]) and may trick the biometric comparison module. However, these activities leave some traces in the extracted features. To such extent, we propose **spoofing detectability**, which indicates the easiness of identifying an ongoing presentation attack that overcame the comparison module and available defenses. A categorization into Low (L), Medium (M), High (H) categories from [36] can be obtained as:

$$\{L, M, H\} = SpoofingDetactability(ID, met, rf, thrLM, thrMH)$$

We first choose a set of intrusion detectors $ID = \{id_1, id_2, ..., id_N\}$ that will be trained on a (labelled) validation set to collect metric scores $M = \{m_1, m_2 \ ... \ m_N\}$ according to a metric *met,* e.g., False Positives, False Negatives, or others as in Section 4.5. Individual scores in M are aggregated through a *reference function rf* into a unique reference value $rv = rf(M)$. Examples of *rf* are average, median, standard deviation of individual scores, or considering only the algorithm that resulted in the best metric score. Once *rv* is computed, we identify two thresholds *thrLM* and *thrMH* to respectively separate Low (L) from Medium (M) and M from High (H) categories. Poor *rv* values will lead to L spoofing detectability. When attacks are identified precisely (i.e., *rv* is almost ideal), spoofing detectability results in the H category.

Circumvention refers to [36] the easiness of fooling the authentication system, and is therefore directly bound to biometric templates and to the comparison module involved in the authentication process. On the other hand, spoofing detectability refers to the ability of suspecting an ongoing attack, independently from

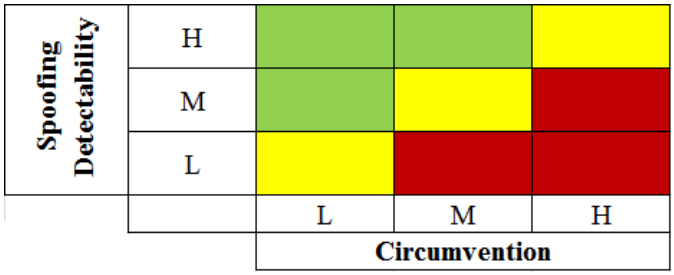

Figure 1. Synergy of Spoofing Detectability / Circumvention. Top-left is the desired, while bottom right should be avoided.

templates and comparison. If a given biometric characteristic can easily be circumvented, but spoofing detectability is sufficiently high, the malicious activity is likely to be identified from the anomaly detector. Therefore, even if the attack fools the biometric comparison, a spoofing detector would realise that an attack is ongoing and thus provide this information to the decision module. As a result, the authentication system may consider the user as non-legitimate. Figure 1 depicts combinations of both spoofing detectability and circumvention, with a green-yellow-red scale highlighting combinations from the most desirable to the worst.

Considering a (multi-)biometric authentication system, the contribution of spoofing detectability mainly concerns the two aspects below.

- Design-time. Spoofing detectability builds an additional property with respect to the properties in Section 2.2; hence its introduction can help selecting the appropriate biometric characteristic(s) for a given system (see Section 6).
- Runtime. During system operation, spoofing detection can complement biometric comparison. As described in Section 7, once a sensor has acquired the biometric sample and the designated module has extracted the features, the latter are sent both to the comparison module and to the spoofing detection module, which operate independently. Results of the two modules become inputs to the decision module, thus they contribute to the final decision about user legitimacy.

4. Experimental Campaign

Our experimental campaign applies the methodology described in the following sub-section. Such methodology requires: datasets containing biometric features and attacks according to a comprehensive attack (sensor spoofing) model described in Section 4.2. Then, we report on unsupervised anomaly detection algorithms in Section 4.3, leaving Section 4.4 to report on, experimental setup, metric(s) and supporting tools.

4.1. Methodology to Execute Experiments

The experiments to substantiate our analysis have been structured according to the following steps:

- **Preparation**. Formatting selected datasets in order to standardize/normalize their characteristics, removing textual features and features which have many missing values.
- **Cropping**. With large datasets, processing may require an unreasonable amount of resources.
- **Injection**. We update the datasets injecting the effect of spoofing attacks on data according to our attack model. Further details are provided in Section 4.3.
- **Splitting Datasets**. For each dataset, we create 2 different files, one to be used for feature selection and training, the other for validation.
- **Experiments**. Selected algorithms are exercised through the RELOAD tool on each of the datasets separately, providing results as triples <*dataset, algorithm, metric value*>.
- **Data Analysis.** Metric values, as well as additional metadata e.g., details of the feature selection process, are aggregated in order to highlight the main findings.

4.2. Selection of Biometric Characteristics and Datasets

We focused on publicly available datasets, shared without constraints except sources referencing. We disregarded datasets containing non-textual information, such as images or audio tracks, which require extracting textual features. The only exception has been fingerprints [25], where we processed the images by using state-of-the-art feature extractors [26]. As shown in Table I, our extensive research process produced 10 datasets related to 8 different biometric characteristics. The datasets include features pertaining to: Fingerprint, Voice, Face, Heart Rate Variability (HRV), Electro Dermal Activity (EDA), human gait (activity recognition), Keystroke, and Hand Gesture.

Table I: Selected Biometric characteristics and Datasets

Biometric Characteristic	Dataset(s)	Dataset(s) Description	# Features	# Data Points
Fingerprint	[25]	CASIA-FingerprintV5 data was captured using URU4000 fingerprint sensor. The volunteers of CASIA-FingerprintV5 contributed 40 fingerprint images of their eight fingers. Images were elaborated by authors of this paper using [26].	15	20 000
Face	[24]	The data set was provided by Dr. Yoshua Bengio of the University of Montreal. It contains 7049 facial images and up to (not always) 15 marked keypoints.	30	7 049
Keystroke	Tappy [18]	Contains keystroke logs collected from over 200 subjects, with and without Parkinson's Disease (PD), as they typed normally on their own computer having installed a custom recording app, Tappy	6	1.5M (appr.)
Heart Rate Variability	SWELL [19], WESAD [20]	(SWELL) The dataset was collected in an experiment in which 25 people performed office work. Creators manipulated working conditions with stressors as email interruptions and time pressure.	66, 62	391 638, 135 650
Human Gait	[22]	A comprehensive gait database of 93 human subjects who walked between two endpoints. Gait data is recorded using two smartphones (right thigh and left side of waist). Additional meta data of an individual is recorded	70	350k (appr.)
Electro Dermal Activity	SWELL [19], WESAD [20]	(WESAD) This multimodal dataset features physiological and motion data, recorded from both a wrist and chest-worn device, 15 subjects during a lab study.	54, 95	98 486, 33 948
Voice	[23]	The dataset is constructed using recorded samples of male and female voices, speech, and utterances. The samples are processed using acoustic analysis tools.	20	3 168
Hand Gesture	Kinect LeapMotion [21]	The dataset contains several different gestures acquired with both the Leap Motion and the Kinect devices, thus allowing the construction and evaluation of hybrid gesture recognition systems.	95	1 400

4.3. Spoofing Attack Model

To quantify the capabilities of anomaly detection algorithms in detecting spoofing attacks [30], [31], [29], [34], [32], [38], we need to i) devise an attack model, and then ii) inject effects of attacks into data. To make this process suitable for all the selected biometric characteristics, we devised an attack model that focuses on alterations of sensor data rather than on alterations related to subsequent phases of the authentication process, as i.e., comparison.

Table II reports on spoofing attacks in [29], [30], [38]. Attacks are aggregated in the table if they share a similar effect on the values of biometric features. For example, reuse of residuals and replay attack will result in resubmitting biometric data which was already presented in the past, or rather providing exact same feature values with respect to a past data point. This allows identifying 4 different categories of effects, namely: *Missing, Reuse, Slight Change, and Multiple Slight Changes*.

The *Missing* category groups threats to availability as Denial of Service. The *Reuse* category aggregates threats where the attacker re-submits data already and legitimately submitted at a previous stage, e.g., reuse of residuals and replay attacks. In the *Slight Change* category, we include many spoofing attacks that forge biometric characteristics producing samples other than the legitimate ones, and similar enough to circumvent the comparison module. The effects of these attacks on data include slight variations of features values, making this category of attacks, among others, the hardest to identify. Finally, the *Multiple Slight Changes* category includes attacks (such as brute-force) that forge many biometric samples in short time span.

Datasets selected for this study do not include attacks data; therefore, we simulate such attacks through fault (attack) injection. Briefly, we inject the categories of attacks in Table II in each selected dataset. The injection is activated randomly, with a probability of 5%, and updates feature values

Table II: Attack Model and Effects of Attacks on Data

Attack Models			Description	Effect on Values	Effect Name
[38] (2018)	[29] (2009)	[30] (2007)			
-	Denial of Service	Denial of Service	The sensor cannot deliver actionable data to the authentication system	Sensor delivers either a default value e.g., 0, or 'no value'	Missing
Class II (Replay)	Reuse of Residuals, Replay Attack	Reuse of Residuals, Replay Attack	The attacker uses past data to deliver (multiple) forged characteristic to the sensor	Past data is sent again to the sensor for three times in a row.	Reuse
Class I (Spoof), Class IV (Replace Values), Class VII (MiM)	Spoofing, MiM, Eavesdropping, Override Feature Extraction	Fake Physical Biometric, Fake Digital Biometric, False Data Inject, Override Feature Extraction	The biometric characteristic is forged to circumvent the authentication system.	The data read from the sensor is altered through statistical operations e.g., belonging to a normal distribution with average and variance related to the last *n* characteristic data.	Slight Change
-	Brute Force	-	A huge amount of slightly different characteristic data is delivered to the system in a short time span.	*m* data points are generated according through statistical operations	Multiple Slight Changes
Class III, (Template Substitution), V (Replace Matcher), VI (Modify DB), VIII (Override)	*Out-of-scope* e.g., Component replacement, Hill Climbing, and *Characteristic-specific attacks*	Latent Print Reactivation, False Enrollment, Synthetized Feature Vector, *Threat Vectors 3.13 – 3.21 [30]*.	These attacks are listed in [38], [29], [30] as attacks to the biometric system, but they are either i) specific for the biometric characteristic, or ii) not related to the presentation of the biometric characteristic and/or the feature extraction process e.g., related to the comparison module and/or template DB, and therefore are discarded in our analysis.		*N/A*

(and, eventually, injecting additional data points) with the effect that is reported in the last column of the table. More in detail, injecting a *Missing* attack forces some feature values to 0, or null e.g., *Keystroke's flightTime*, which is usually microseconds, may set to 0. Instead, injecting a *Reuse* repeats a single data point which already appeared in the recent past (randomly chosen amongst the last 20 data points). To inject a *Slight Change*, we calculate average and standard deviation for each feature by using the 100 data points previous to the injection to define a context. Then we randomly sample some feature values in the range defined by the confidence interval *average* ± *standard deviation*. For Multiple Slight Change, this injection is repeated by adding *n* rows to the dataset, with *n* randomly chosen in the interval [2, 5]. Note that all attacks but this last category inject a new row in the dataset, simulating the forgery of the biometric trait for spoofing purposes. After the injection process, the rate between normal data points and anomalies due to attacks in the datasets is around 8%.

4.4. Unsupervised Anomaly Detectors

We employ a set of 9 unsupervised algorithms to estimate the detectability of spoofing attacks to biometric characteristics. First, we select a well-known algorithm for each family in Section 2.1 as follows. We identify the variant [15] for binary classification of Support Vector Machines (*One-Class SVM*) and *DBSCAN* [46] clustering algorithm. Regarding angle and neighbor-based families we select the Outlier Detection using Indegree Number (*ODIN*, [14]) and the Angle-Based Outlier Detection (*ABOD*, [13]) algorithms, along with the more recent Histogram-Based Outlier Score (*HBOS*, [12]) and Sparse Data Observers (*SDO*, [47]) algorithms. Unfortunately, we had to discard ABOD due to its high computational complexity (cubic), re-directing our choice to *FastABOD* [13], which scales down the complexity through a nearest-neighbour search.

Moreover, since algorithms may have semantic overlaps among families, (as happens with FastABOD), in the second phase of the selection process we choose other 3 algorithms which have cross-cutting peculiarities. Neighbours identification is employed to reduce noise and computational

complexity in the stochastic *ISOS* [45], and in the density-based *COF* [44]. Ultimately, we consider *LDCOF* [43], which builds a density-based anomaly detector using an internal clustering procedure.

4.5. Experimental Setup, Metrics and Tools

We describe here the experimental setup for our study. We downloaded the datasets in Section 4.2 from their repositories shaping them as CSV files. Then, we downloaded the latest release of RELOAD [52], a tool for evaluation of unsupervised anomaly detectors that is publicly available on GitHub. We used MCC [1] as target metric to evaluate detection capability of algorithms as it fits also unbalanced datasets [5], which often happens in the security domain i.e., many normal data and only a few attacks. Then, we select the bet 10 features of each dataset according to their information gain [27]. We also proceeded with a 10-fold sampling of the training set [28]. Metrics [2] other than MCC are reported for completeness and comparison with the state of the art. We have run experimental campaigns including all the datasets and algorithms considered in this study. The experiments have been executed on a server equipped with Intel Core i7-6700 with four 3.40GHz cores, 24GB of RAM and 100GB of user storage. Overall, executing the experiments reported in this paper, required approximately one month of 24H execution. All the metric scores and files that we used to collect and summarize values are publicly available at [53].

5. Results and Discussion

This section describes and comments on the results of our experimental campaign with the aid of Table III. The table shows, for each dataset, the best algorithm(s), and the metric scores.

For several datasets - namely EDA(SWELL), HRV(WESAD) and Voice – multiple algorithms provided the same detection scores. Despite our selection of heterogeneous algorithms, sometimes algorithms make the same choice, resulting in very similar, if not exactly the same, detection scores. From a general standpoint, accuracy scores achieved by the best algorithm in each dataset always exceed 95%: this indicates that only less than 5% of the biometric samples are being misclassified, either as a False Positive (FP - benign sample interpreted as a tentative of spoofing attack) or as a False Negative (FN - spoofing attack not detected). Additionally, we can observe how Precision scores are overall higher than Recall scores. This indicates that most of the misclassifications are FNs, or rather spoofing attacks that were not detected, representing a potential harm to the system.

Algorithms which appear in the second column of Table III in the majority of the cases are COF – datasets EDA(SWELL), Face, Hand Gesture, HRV(WESAD) – and ODIN, which takes the lead on Fingerprint, HRV(SWELL), Human Gait and Keystroke datasets. Both algorithms are based on nearest-neighbors as well as FastABOD, which shows the best detection scores for the EDA(WESAD). This highlights that for 9 datasets out of the 10, the best algorithm embeds a nearest-neighbor search. Our dataset sample is not big enough to prove that this trend is valid in general for

Table III: Best Algorithms (and used features) for each dataset, along with Metric Scores.

Dataset	Best Algorithm	FPR	Precision	Recall	FMeasure	MCC	Accuracy	AUC
EDA (SWELL)	DBSCAN, COF, ISOS, SVM, LDCOF, SDO	0.52	89.11	49.39	63.10	63.93	95.00	74.43
EDA (WESAD)	FASTABOD	0.00	100.00	34.98	50.18	56.45	94.70	67.49
Face	COF	1.29	76.67	57.95	65.54	64.29	95.70	78.33
Fingerprint	ODIN	0.00	100.00	66.80	78.79	78.85	96.55	83.40
Hand Gesture	COF	1.63	76.92	62.50	68.97	66.98	95.50	80.43
HRV (SWELL)	ODIN	0.00	100.00	62.15	75.01	76.42	96.31	81.07
HRV (WESAD)	DBSCAN, COF, ISOS, SVM, LDCOF, SDO	0.55	91.74	60.49	72.91	72.58	95.88	79.97
Human Gait	ODIN	0.54	93.27	98.82	95.88	95.64	99.40	99.14
Keystroke	ODIN	0.00	100.00	60.88	73.93	75.42	95.95	80.44
Voice	DBSCAN, SVM, ISOS, LDCOF, SDO	0.55	89.39	54.17	67.16	67.29	95.25	76.81

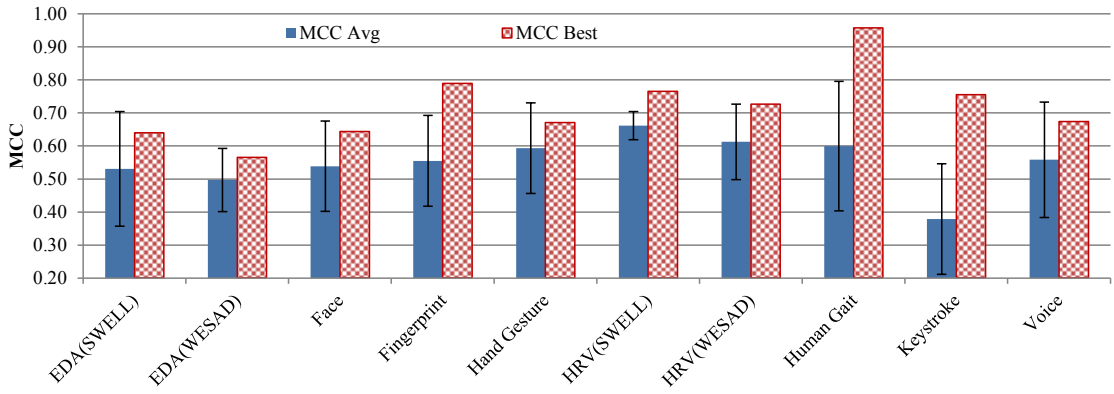

Figure 2. Average MCC of algorithms (with st.dev) and MCC score of the best algorithm (Table IV) on each dataset.

most of the existing biometric characteristics. However, it allows focusing on these algorithms to apply anomaly detection to detect presentation / spoofing attacks within the datasets considered in this study.

To conclude the presentation and the discussion of experimental results, Figure 2 reports a chart where: i) blue solid bars correspond to average MCC scores of algorithms (with std-based error bars), while the red pattern-filled columns represent the MCC score of the best algorithm for each dataset. In some cases, the difference between the two data series is notable: for fingerprint, keystroke and human gait the optimal algorithm results in an MCC score that is more than 1.5 times higher than the average. While this aspect is not surprising, it further remarks how the choice of the algorithm really affects the system: although the selection of the correct algorithms for a given system or dataset is currently under investigation, [10], [55], [51] there are no clear answers to this problem (yet?).

6. Quantification of Spoofing Detectability

To calculate spoofing detectability we need to instantiate (see Section 3) the parameters *ID, met, rf, thrLM, thrMH*. The set of intrusion detectors *ID = {DBSCAN, HBOS, ODIN, FastABOD, SVM, SDO, ISOS, COF, LDCOF}* includes all the algorithms selected in this study, while *met = MCC*. To provide a complete and solid view on spoofing detectability of biometric properties, we instantiate two functions *SDAvg* and *SDBest: SDAvg* considers the average of MCC scores as *rf*, while *SDBest* works with maximum absolute value of MCC as reference function.

SDAvg(ID, met, rf="average", thrLM=55.3, thrMH=71.8)

SDBest(ID, met, rf="max_abs" , thrLM=65.0, thrMH=80.0)

For each function SDAvg and SDBest, *thrLM* and *thrMH* were arbitrarily chosen to balance results; in fact, these thresholds make at least one biometric characteristic fall in each category L, M, H. We are aware that assigning values to these thresholds heavily affects the outcome of these functions. This study wants to provide a general view on spoofing detectability without domain specific-constraints. For example, in some domains lowering FNs has priority with respect to minimizing FPs, and therefore *met* other than MCC e.g., FScore with $\beta > 1$, should be chosen, while thresholds *thrLM, thrMH* need to be tuned accordingly.

Table IV shows the outcome of our spoofing detectability property. For each *Dataset* (first column), we report: the *Biometric Characteristic*, the average *MCC* of algorithms for a given dataset and *MCC* of the *Best* algorithm on such dataset, the results of *Spoofing Detectability,* and *Circumvention* [36]. *"Final"* Spoofing Detectability is obtained as a combination of SDAvg and SDBest results. If individual results agree, final category is obtained straighforwardly; when different, indvidual results are merged by majority. As a tie-breaker, we looked at MCC scores to decide on the final category. Consequently, face resulted in M and L individual scores, with very similar scores to EDA – especially when considering the SWELL dataset -, and therefore we set the final category value of Face as EDA's. As a side remark, in many cases the categories obtained by looking at average and best MCC hold, i.e., columns *SDAvg*, *SDBest* are the same for half of the datasets.

Table IV: Overview of Spoofing Detectability for the considered datasets and biometric characteristics.

Dataset	Biometric Characteristic	MCC		Spoofing Detectability			Circumvention
		Avg	Best	SD Avg	SD Best	Final	[36]
SWELL	**EDA**	53.1	63.9	M	L	**L**	**L**
WESAD		49.7	56.5	L	L		
Face	**Face**	53.9	64.3	M	L	**L**	**H**
Fingerprint	**Fingerprint**	55.5	78.9	M	M	**M**	**M**
Hand Gesture	**Hand Gesture**	59.3	67.0	M	M	**M**	**M**
SWELL	**HRV**	66.1	76.4	H	M	**M**	**L**
WESAD		61.2	72.6	M	M		
Human Gait	**Human Gait**	59.9	95.6	M	H	**H**	**M**
Keystroke	**Keystroke**	37.9	75.4	L	M	**M**	**M**
Voice	**Voice**	55.8	67.3	M	M	**M**	**H**

Looking at Table IV, we can notice how EDA and Face characteristics resulted in L values of spoofing detectability, meaning that may not be trivial to detect attacks directed to the related sensors. The best anomaly detection algorithms will still often make mistakes (i.e., MCC values are lower than 65%) and, more importantly, they will not detect more than 60% of the attacks (see Recall column in Table III). A completely different scenario is exhibited by Keystroke, which has the lowest average MCC scores of 37.9, which almost doubles when considering the best algorithm. This also motivates the M value for spoofing detectability that was assigned to such biometric characteristic, despite it showed the lowest average MCC score. Instead, out of the 8 biometric characteristics considered in this study, only Human Gait was categorized as H spoofing detectability, mainly due to the almost perfect detection capabilities that ODIN – again – showed in detecting spoofing attacks in this particular dataset.

7. Runtime Spoofing Detectability

Spoofing detectability was primarily meant to be a property of biometric characteristics to be used at design-time of a system: however, we show here how to setup a runtime support for the final decision about authentication.

7.1. General Architecture

As shown in Figure 3, typical biometric authentication systems require the user a biometric sample that is acquired by sensors. Then, feature values are extracted and delivered to the comparison module, which computes a score that enables the system to decide on authentication. A Spoofing Detection module can work in parallel with respect to the comparison module, relying on the same inputs but aiming at detecting suspicious feature sets instead than comparing feature values with biometric templates. The Spoofing Detection module runs an anomaly detection algorithm trained or

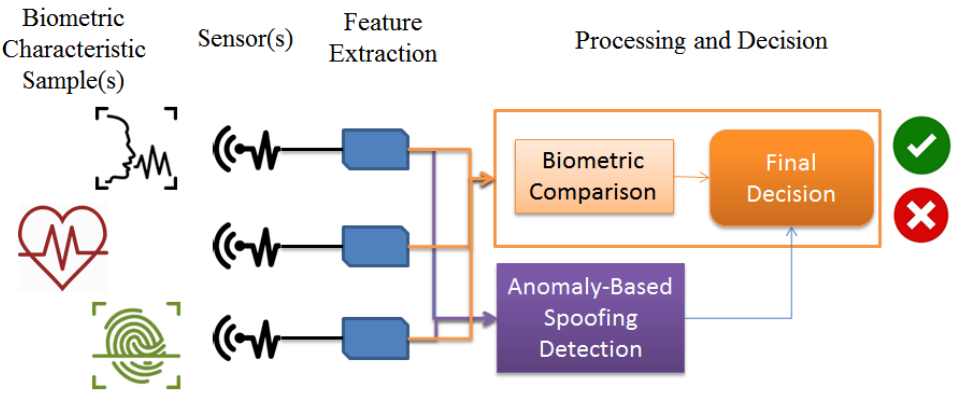

Figure 3. General Runtime support offered by Spoofing Detectability: Spoofing Detection module.

updated when acquiring the biometric templates and uses the model learned during training to decide on anomalies at runtime. This result, alongside with the score produced by the comparison module, is sent to the Final Decision module, which accepts or rejects the user depending on those inputs.

7.2. Case Study: ProtectID Project

An instantiation of the architecture described above was designed – and is currently under implementation – in the scope of the *ProtectID* [54] project. One of the goals of this project is to design and implement a cyber-physical system that allows citizens to interact with remote services offered by public administration through commercial devices. The access to their personal data and other sensitive information raises privacy issues [50] that are mitigated through robust (biometric) authentication strategies. Amongst all the candidate characteristics, project partners, together with the public administrations, selected *Fingerprint*, *Face* and *Keystroke* due to the high availability of related sensors in personal devices. In this context, spoofing detectability could not help in selecting biometric characteristics; however, the introduction of a spoofing detection module can provide runtime support to the recognition of the above-mentioned characteristics, and corroborate or overturn the final decision. A high-level view of *ProtectID* system is reported in Figure 4. Briefly, if citizens want to use some service provided by public administration, they 1) connect to the web portal. This action generates an authentication request which goes 2) through the authentication server and 3) reaches the citizen device. Now, 4) the user has to provide biometric samples for authentication, which are processed for feature extraction, encrypted and 5) sent back to the server; the latter 6) processes the features extracted from the sample for authentication. Finally, if the authentication succeeds, 7) a token is generated to 8) let the citizen access the system and, eventually, 9) take advantage of the service(s).

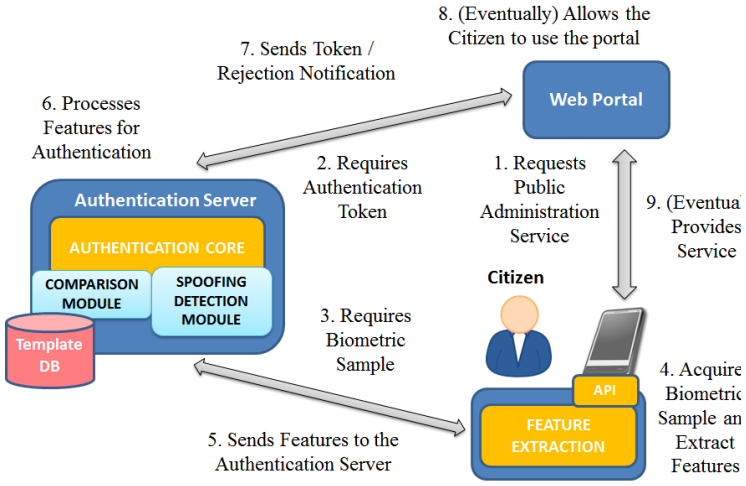

Figure 4. Spoofing Detection in *ProtectID* project.

7.3. Spoofing Detectability in ProtectID

While an extensive description and discussion of the authentication system of *ProtectID* project is out of the scope of this paper, we point out here the role of spoofing detectability. In particular, the Authentication Server of *ProtectID* project (see left-side of Figure 4) was equipped with a *Spoofing Detection Module*, which complements the biometric comparison module, providing an indication of the trustworthiness of the set of feature values obtained from the biometric sample. The Comparison Output *co* and the Spoofing Detection Output *sdo*, along with the Spoofing Detectability categories' values of the *Fingerprint (M)*, *Face (L)* and *Keystroke (M)*, are used to grant authentication as:

$$Authentication = Final\ Decision(co, sdo, <M, L, M>)$$

The *Final_Decision* function to be implemented in *ProtectID* is confidential yet. However, the above formula aims at showing the runtime applicability of our solution, but it should be instantiated to suit the specific target system. As a last remark, we want to point out a side effect of our experimental study on spoofing detection of biometric characteristics. As reported in Table III, the ODIN algorithm showed the better performance in detecting spoofing attacks directed to *Fingerprint* and *Keystroke*, performing quite well also for the *Face* characteristic, namely with an MCC of 0.58 (see [53], tab *MainData*) compared to the optimum of 0.64 provided by COF (again, see Table III). Therefore, once

the *ProtectID* system will be developed, its Spoofing Detection Module will rely on the neighbor-based algorithm ODIN to calculate *sdo*.

8. Conclusions

This paper introduced *spoofing detectability,* an additional property of biometric characteristics that categorizes the probability of detecting an ongoing spoofing or presentation attack to overcome available defenses. The purpose of our study is dual: it i) devises an additional property that can help selecting the most appropriate biometric characteristic(s) for a given system, and ii) provides actionable information to define and implement a spoofing detection module that complements the traditional authentication process at runtime. We conducted experiments to quantitatively estimate spoofing detectability. Detection of spoofing attacks to biometric sensors was realized through anomaly detection, an overall solid answer to unknown or zero-day attacks that the attacker may conduct against a biometric authentication system. We selected different unsupervised anomaly detection algorithms, which were then exercised on public datasets related to face, fingerprint, voice, keystroke, heart rate variability, electrodermal activity, human gait, and hand gesture characteristics.

Results of the experimental campaign were presented and discussed, elaborating: i) average detection scores of algorithms, ii) best algorithms to detect presentation / sensor spoofing attacks targeting a given biometric characteristic, and finally devising iii) categories for the spoofing detectability property based on the outcomes of the experimental campaign. Lastly, we show how to define a spoofing detection module as a runtime support to the traditional biometric authentication process for a given system.

Acknowledgments

This work has been partially supported by the Italian PON ProtectID (“Processi e tecnologie innovative per la protezione delle identità digitali e delle informazioni personali in rete”) and by the H2020 programme under the Marie Sklodowska-Curie grant agreement 823788 (ADVANCE) projects. Portions of the research in this paper use the CASIA-FingerprintV5 collected by the Chinese Academy of Sciences' Institute of Automation (CASIA).

References

[1] Boughorbel, Sabri, Fethi Jarray, and Mohammed El-Anbari. "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric." PloS one 12.6 (2017): e0177678.
[2] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” 2011
[3] “Elki data mining,” elki-project.github.io, accessed: 2020-02-20
[4] “Weka 3: Data Mining Software in Java”, www.cs.waikato.ac.nz/~ml/weka/, accessed: 2020-02-20
[5] Chicco, Davide. "Ten quick tips for machine learning in computational biology." BioData mining 10.1 (2017): 35.
[6] McKinney, Wes. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.", 2012.
[7] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the 2003 SIAM Int. Conference on Data Mining, pages 25-36. SIAM, 2003.
[8] Zoppi, T., Ceccarelli, A., Capecchi, T., & Bondavalli, A. (2021). Unsupervised Anomaly Detectors to Detect Intrusions in the Current Threat Landscape. ACM/IMS Transactions on Data Science, 2(2), 1-26.
[9] Chandola, V., Banerjee, A., Kumar, V. “Anomaly detection: A survey”. (2009) ACM computing surveys (CSUR), 41(3), 15.
[10] M.Goldstein and S.Uchida, “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” PloS one, vol. 11,no. 4, p.e 152 - 173, 2016.
[11] He, S., Zhu, J., He, P., & Lyu, M. R. (2016, October). Experience report: system log analysis for anomaly detection. In Software Reliability Engineering (ISSRE), 2016 IEEE 27th International Symposium on (pp. 207-218). IEEE.

[12] Goldstein, Markus, and Andreas Dengel. "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm." KI-2012: Poster and Demo Track (2012): 59-63.
[13] Kriegel H-P, Zimek A. “Angle-based outlier detection in high-dimensional data”. Proc. of the 14th ACM SIGKDD Int. Conference on Knowledge discovery and data mining; ‘08. p. 444–452.
[14] Hautamaki, V., Karkkainen, I., & Franti, P. (2004, August). Outlier detection using k-nearest neighbour graph. In Pattern Recognition. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 3, pp. 430-433). IEEE.
[15] M. Amer, M. Goldstein, and S. Abdennadher, “Enhancing one-class support vector machines for unsupervised anomaly detection,” in Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. ACM, 2013, pp. 8–15.
[16] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In ACM sigmod record (Vol. 29, No. 2, pp. 93-104). ACM.
[17] Schubert, E., Koos, A., Emrich, T., Züfle, A., Schmid, K. A., & Zimek, A. (2015). A framework for clustering uncertain data. Proceedings of the VLDB Endowment, 8(12), 1976-1979.
[18] Adams, Warwick R. "High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing." PloS one 12.11 (2017): e0188226.
[19] Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M., & Kraaij, W. (2014). The SWELL Knowledge Work Dataset for Stress and User Modeling Research. To appear in: Proceedings of the 16th ACM International Conference on Multimodal Interaction (ICMI 2014) (Istanbul, Turkey, 12-16 November 2014)
[20] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, Kristof Van Laerhoven, "Introducing WESAD, a multimodal dataset for Wearable Stress and Affect Detection", ICMI 2018, Boulder, USA, 2018
[21] A. Memo, L. Minto, P. Zanuttigh, "Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition", STAG: Smart Tools & Apps for Graphics, 2015
[22] Vajdi, A., Zaghian, M. R., Farahmand, S., Rastegar, E., Maroofi, K., Jia, S., ... & Bayat, A. (2019). Human Gait Database for Normal Walk Collected by Smart Phone Accelerometer. arXiv preprint arXiv:1905.03109.
[23] Kaggle - Voice Recognition, Jeganathan Kolappan. https://www.kaggle.com/jeganathan/voice-recognition (online), accessed: 2019-11-20
[24] Kaggle - Face Images with Marked Landmark Points, Omri Goldstein (online). https://www.kaggle.com/drgilermo/face-images-with-marked-landmark-points, accessed: 2020-11-20
[25] BIT – Biometrics Ideal Test, CASIA-FingerprintV5, http://biometrics.idealtest.org/
[26] MathWorks - FingerPrint Matching: A simple approach, https://it.mathworks.com/matlabcentral/fileexchange/44369-fingerprint-matching-a-simple-approach (online), accessed: 2019-11-20
[27] Azhagusundari, B., and Antony Selvadoss Thanamani. "Feature selection based on information gain." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2.2 (2013): 18-21.
[28] Moore, Andrew W. "Cross-validation for detecting and preventing overfitting." School of Computer Science Carneigie Mellon University (2001).
[29] Li, Stan Z. Encyclopedia of Biometrics: I-Z. Vol. 2. Springer Science & Business Media, 2009.
[30] Roberts, Chris. "Biometric attack vectors and defences." Computers & Security 26.1 (2007): 14-25.
[31] Gupta, P., Behera, S., Vatsa, M., & Singh, R. (2014, August). On iris spoofing using print attack. In 2014 22nd International Conference on Pattern Recognition (pp. 1681-1686). IEEE.
[32] Biggio, B., Akhtar, Z., Fumera, G., Marcialis, G. L., & Roli, F. (2012). Security evaluation of biometric authentication systems under real spoofing attacks. IET biometrics, 1(1), 11-24.
[33] Chingovska, Ivana, Andre Rabello Dos Anjos, and Sebastien Marcel. "Biometrics evaluation under spoofing attacks." IEEE transactions on Information Forensics and Security 9.12 (2014): 2264-2276.
[34] Nixon, K. A., Aimale, V., & Rowe, R. K. (2008). Spoof detection schemes. In Handbook of biometrics (pp. 403-423). Springer, Boston, MA.
[35] L. Hong, A. K. Jain, S. Pankanti. “Can multibiometrics improve performance?”. In Proceedings AutoID Vol. 99, pp. 59-64, 1999.
[36] A.K.Jain, A. Ross, S. Prabhakar. “An Introduction to Biometric Recognition”. IEEE Transactions on Circuits and Systems for Video Technology, 2004.
[37] A. Azzini, S. Marrara, R. Sassi, F. Scotti. “A fuzzy approach to multimodal biometric continuous authentication”. Fuzzy Optimization and Decision Making, 7(3), 243-256, 2008.
[38] W. Dahea, HS Fadewar. “Multimodal biometric system: A review”, International Journal of Research in Advanced Engineering and Technology, Volume 4; Issue 1; January 2018; Page No. 25-31.
[39] Stallings, W., Brown, L., Bauer, M. D., & Bhattacharjee, A. K. (2012). Computer security: principles and practice (pp. 978-0). Upper Saddle River, NJ, USA: Pearson Education.

[40] E. Schiavone, A. Ceccarelli, A. Carvalho, A. Bondavalli. “Design, implementation, and assessment of a usable multi-biometric continuous authentication system”. In Int. J. Critical Computer-Based Systems, Vol. 9, No. 3, 2019.
[41] Matsumoto, T., Matsumoto, H., Yamada, K., and Hoshino, S. Impact of artificial "gummy" fingers on fingerprint systems. In Proc. of SPIE Opt. Sec. Counterfeit Deterrence Tech. IV, pages 275-289, 2002.
[42] Marasco, E., Shehab, M., & Cukic, B. (2016, October). A Methodology for Prevention of Biometric Presentation Attacks. In 2016 Seventh Latin-American Symposium on Dependable Computing (LADC) (pp. 9-14). IEEE.
[43] Mennatallah Amer and Markus Goldstein. 2012. Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In Conference: Proceedings of the 3rd RapidMiner Community Meeting and Conferernce (RCOMM 2012).
[44] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. 2002. Enhancing effctiveness of outlier detections for low density patterns. In Pacifi-Asia Conference on Knowledge Discovery and Data Mining. Springer, 535–548.
[45] Schubert, E., & Gertz, M. (2017, October). Intrinsic t-stochastic neighbor embedding for visualization and outlier detection. In International Conference on Similarity Search and Applications (pp. 188-203). Springer, Cham.
[46] Martin Ester,Han-peter Kriegel,Jorg Sander, Xiaowei Xu,”A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, 2nd International conference on Knowledge Discovery and Data Mining (KDD-96)
[47] Vázquez, Félix Iglesias, Tanja Zseby, and Arthur Zimek. "Outlier detection based on low density models." 2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2018.
[48] Saini, Rupinder, and Narinder Rana. "Comparison of various biometric methods." International Journal of Advances in Science and Technology 2.1 (2014): 2.
[49] Srivastava, H. (2013). A Comparison Based Study on Biometrics for Human Recognition. Journal of Computer Engineering (IOSR-JCE), 15(1), 22-29.
[50] Voigt, P., & Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing.
[51] Zoppi, Tommaso, Andrea Ceccarelli, and Andrea Bondavalli. "MADneSs: a Multi-layer Anomaly Detection Framework for Complex Dynamic Systems." IEEE Transactions on Dependable and Secure computing (2019).
[52] Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2019). “Evaluation of Anomaly Detection algorithms made easy with RELOAD”. In Proc. of the 30th Int. Symposium on Software Reliability Engineering (ISSRE), pp 446-455, IEEE.
[53] Data about experimental campaign (online) *https://drive.google.com/file/d/1d3s6eaXmgD3LEC_JTspPqaaTUVWFCKqd/view?usp=sharing* (accessed: 2021-03-04)
[54] *ProtectID website* (online), https://www.protectid.it/ accessed: 2021-02-20
[55] Zoppi, T., Ceccarelli, A., Salani, L., & Bondavalli, A. (2020). On the educated selection of unsupervised algorithms via attacks and anomaly classes. Journal of Information Security and Applications, 52, 102474.
[56] Bowyer, K. W. (2004). Face recognition technology: security versus privacy. IEEE Technology and society magazine, 23(1), 9-19.
[57] D. Menotti et al., Deep representations for iris, face, and fingerprint spoofing detection, IEEE Trans. Inform. Forensics Sec. 10 (4) (2015) 864–879.
[58] Sajjad, Muhammad, Salman Khan, Tanveer Hussain, Khan Muhammad, Arun Kumar Sangaiah, Aniello Castiglione, Christian Esposito, and Sung Wook Baik. "CNN-based anti-spoofing two-tier multi-factor authentication system." Pattern Recognition Letters 126 (2019): 123-131.
[59] Chaos Computer Club Berlin. (2013). Hacking iPhone 5S Touchid, YouTube. [Online]. www.youtube.com/watch?v=HM8b8d8kSNQ
[60] The CNN. (2010). Man in Disguise Boards International Flight. [Online]. edition.cnn.com/2010/WORLD/americas/11/04/canada.disguised.passenger/
[61] Spoof attacks top this week biometrics and digital ID news (2019) [Online] biometricupdate.com/201911/spoof-attacks-top-this-weeks-biometrics-and-digital-id-news
[62] Schuckers, S. A. (2002). Spoofing and anti-spoofing measures. Information Security technical report, 7(4), 56-62
[63] Arashloo, Shervin Rahimzadeh, Josef Kittler, and William Christmas. "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol." IEEE access 5 (2017): 13868-13882.