Revisiting the Security of Biometric Authentication Systems Against Statistical Attacks

SOHAIL HABIB, HASSAN KHAN, and ANDREW HAMILTON-WRIGHT,
University of Guelph
URS HENGARTNER, University of Waterloo

The uniqueness of behavioral biometrics (e.g., voice or keystroke patterns) has been challenged by recent works. Statistical attacks have been proposed that infer general population statistics and target behavioral biometrics against a particular victim. We show that despite their success, these approaches require several attempts for successful attacks against different biometrics due to the different nature of overlap in users' behavior for these biometrics. Furthermore, no mechanism has been proposed to date that detects statistical attacks. In this work, we propose a new hypervolumes-based statistical attack and show that unlike existing methods, it (1) is successful against a variety of biometrics, (2) is successful against more users, and (3) requires fewest attempts for successful attacks. More specifically, across five diverse biometrics, for the first attempt, on average our attack is 18 percentage points more successful than the second best (37% vs. 19%). Similarly, for the fifth attack attempt, on average our attack is 18 percentage points more successful than the second best (67% vs. 49%). We propose and evaluate a mechanism that can detect the more devastating statistical attacks. False rejects in biometric systems are common, and by distinguishing statistical attacks from false rejects, our defense improves usability and security. The evaluation of the proposed detection mechanism shows its ability to detect on average 94% of the tested statistical attacks with an average probability of 3% to detect false rejects as a statistical attack. Given the serious threat posed by statistical attacks to biometrics that are used today (e.g., voice), our work highlights the need for defending against these attacks.

CCS Concepts: • **Security and privacy** → **Biometrics;**

Additional Key Words and Phrases: Statistical attacks, behavioral biometrics, voice authentication, gait authentication, keystroke authentication

ACM Reference format:
Sohail Habib, Hassan Khan, Andrew Hamilton-Wright, and Urs Hengartner. 2023. Revisiting the Security of Biometric Authentication Systems Against Statistical Attacks. *ACM Trans. Priv. Sec.* 26, 2, Article 21 (April 2023), 30 pages.
https://doi.org/10.1145/3571743

This work was supported by the NSERC through grant RGPIN-2019-05120.

Authors' addresses: S. Habib, H. Khan, and A. Hamilton-Wright, School of Computer Science, University of Guelph, Canada; emails: {shabib03, hassan.khan, andrew.hamilton-wright}@uoguelph.ca; U. Hengartner, Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada; email: urs.hengartner@uwaterloo.ca.

2471-2566/2023/04-ART21 $15.00
https://doi.org/10.1145/3571743

1 INTRODUCTION

Researchers have shown that a variety of biometrics are unique enough to warrant identity verification of users. These biometrics include keystroke [Bergadano et al. 2002; Monrose and Rubin 1997], voice [Chung et al. 2018; Nagrani et al. 2017], mouse movement [Ahmed and Traore 2007; Jorgensen and Yu 2011], stylometry [Brocardo et al. 2014; Fridman et al. 2016], gait [Boulgouris et al. 2005; Liu and Sarkar 2006], gaze [Liu et al. 2015; Maeder et al. 2004], and touch input behaviour [Frank et al. 2013; Li et al. 2013]. Financial institutions and enterprises are increasingly adopting biometrics to corroborate their customers' and employees' identities [Lee 2016; Mayhew 2016]. A few banks have provided the option of voice authentication through Amazon Alexa to their customers [Burt 2018]. Several commercial solutions (including Samsung's Nexsign) enable identity verification through behavioral biometrics on personal devices [BehavioSec 2021; Samsung SDS 2021].

Some researchers have revisited the claimed security of various biometrics. Biometrics authentication systems rely on machine learning and statistical analysis techniques to model the normal behavior of a user and then flag anomalies by identifying deviations from normal behavior. Researchers have proposed attacks that exploit the overlaps in users' behavior across the population ("statistical attacks") [Negi et al. 2018; Serwadda and Phoha 2013a, b; Serwadda et al. 2011]. To mount a statistical attack, the attacker uses behavioral data of the population (a subset of users excluding the victim) to infer the "normal behaviors" of the population and uses these to defeat the biometrics-based authentication system against the victim. Although less successful than other attack approaches [Khan et al. 2016, 2018; Tey et al. 2013], statistical attacks are less expensive to set up since they do not require any knowledge of the behavior of the victim.

Previous works have devised different methods to infer general population statistics [Negi et al. 2018; Serwadda and Phoha 2013a, b; Serwadda et al. 2011; Zhu et al. 2020]. However, these approaches do not quantify the degree of overlap between the regions capturing the behavior associated with users, which helps identify the most promising attack samples. Furthermore, existing approaches work well when most of the population is closer to the population mean. If the population is distributed within clusters, there may be multiple overlapping regions and the attacker needs to navigate the entire cluster space for victims that are close to the centroid of the clusters. Consequently, several existing approaches [Serwadda and Phoha 2013a, b; Serwadda et al. 2011] require a large number of failed attempts before success for a significant proportion of their victims. For a targeted attack (i.e., where an attacker targets a particular victim), it is desirable for the attacker to have fewest failed attempts possible to avoid getting locked out or raising an alarm [Acar et al. 2020; Khan et al. 2018; Negi et al. 2018]. Finally, no mechanism has been proposed to detect statistical attacks on biometrics.

We propose to use hypervolumes to identify overlapping regions in the biometrics space. Hypervolumes have previously been explored in ecology and evolutionary studies to describe and compare niches or trait spaces characterizing phenotypes [Hutchinson 1957]. We then use the overlapping regions to infer population statistics and mount statistical attacks. Our intuition for using hypervolumes comes from the fact that data used to capture users' behavior is multidimensional and can be modeled as an n-dimensional hypervolume, allowing us to calculate the overlapping regions. The identified overlapping regions will result in uncertain boundaries for classification, which could lead to misclassifications. For adversaries, the overlapping regions are of interest since they may attempt to impersonate the victim by submitting samples that come from the overlapping regions of the population, thereby increasing the probability to mount a successful attack. Previous statistical attacks have considered population means [Negi et al. 2018; Serwadda and Phoha 2013b; Serwadda et al. 2011] or better feature space traversing approaches [Negi et al. 2018] to exploit this overlap (see Figure 2); however, our approach determines precise overlap regions using hypervolumes (see Figure 2), thereby significantly improving attack performance by reducing the search space and extracting the statistics from overlapping regions where the boundaries for classification would be uncertain. Furthermore, unlike previous approaches that assume a unimodal data distribution by completely relying on the population mean [Serwadda and Phoha 2013b; Serwadda et al. 2011], hypervolumes work for multimodal distributions too.

To demonstrate the efficacy of our attack, we compare it with three existing statistical attack approaches [Negi et al. 2018; Serwadda and Phoha 2013b; Serwadda et al. 2011]. We evaluate our approach by attacking five diverse biometrics: touch input [Frank et al. 2013], mouse movement [Eberz et al. 2018], gait [Zou et al. 2020], keystroke dynamics [Killourhy and Maxion 2009], and voice [Chung et al. 2018; Nagrani et al. 2017; Zhao et al. 2020]. We evaluate these attacks on two metrics—the percentage of compromised population and number of attempts required for a successful attack. We show that our attack performs consistently well in all tested scenarios. On the first attempt, it outperforms the second-best attack, on average, by 18 percentage points (37% vs. 19%). It consistently performs better against most scenarios, and after 10 attempts, on average, it compromises at least 17 percentage points (77% vs. 60%) more victims than the second-best attack for different biometrics.

Some researchers have proposed to use biometrics in a multimodal fashion to circumvent statistical attacks [Acar et al. 2020; Stanciu et al. 2016]; however, we propose the first ever mechanism to detect statistical attacks. False rejects in biometrics systems are common. Distinguishing failed statistical attack attempts from users' samples ensures the usability of the authentication biometric by not locking out legitimate users unnecessarily. Our detection mechanism exploits the attacker's desire to efficiently break the system against a particular victim with as few attempts as possible. This assumption has been made by other researchers in this area [Acar et al. 2020; Khan et al. 2018; Negi et al. 2018], as a large number of failed attempts within a short interval can lock out the victim's account and spreading the attack over a longer period can take much longer. Unlike a legitimate user, when a sample submitted by the statistical attacker gets rejected, the attacker submits a new sample different from the rejected one and different from users' previously observed behavior with non-trivial probability. When presented with two consecutive rejected samples, our detection mechanism considers the classifier's decision, the distance of the samples from the centroid of the victim's samples, and the distance between the samples to determine whether they are from a statistical attack source. We validate our detection mechanism against the attacks evaluated in this work. We evaluate our detection mechanism against users for whom statistical attacks are not successful in the first attempt. Our evaluation shows that our detection mechanism is, on average, able to detect 94% samples of the best-performing attacks (Hypervolume and K-means++) while misclassifying, on average, only 3% of users' samples. Our main contributions include the following:

- A novel statistical attack method that outperforms existing statistical attacks in terms of the number of users it can successfully attack and the number of failed attempts before a successful attack. This superior performance of our attack is demonstrated against five diverse biometrics with four different classifiers.
- First-ever demonstration of the susceptibility of the voice biometric to a statistical attack.
- A detection mechanism against statistical attacks that can reliably detect statistical attacks with low **False Detection Rates (FDRs)** using as few as two samples.
- An open source release of our proposed attack (evaluated on five public datasets) and detection mechanism for the research community to reproduce our results and advance research in this area.[1]

[1] https://github.com/sohailhabib/SecurityMetrics.

2 BACKGROUND AND RELATED WORK

In this section, we first outline how biometrics are used for user authentication and metrics used by the proposals and attack work. We then discuss three broad categories of attacks on biometrics-based authentication systems.

User authentication using biometrics has been modeled as a binary classification problem—both by original proposals [Frank et al. 2013; Xu et al. 2014] or attacks on these proposals [Serwadda et al. 2011, 2016]. We note that user authentication using biometrics is sometimes also modeled as an anomaly-detection problem [Killourhy and Maxion 2009]. Since statistical attacks attempt to submit an attack sample that is possibly close to the users' samples, previous works have successfully targeted both binary classification [Serwadda and Phoha 2013b] and anomaly detection [Serwadda et al. 2011] approaches. In the binary classification problem, the training data constitutes of a positive class and a negative class. The positive class comprises of a subset of data of the user, and the negative class comprises of data from multiple other users. The positive test cases are the remaining samples from the target user, whereas the negative test cases are samples from other users not considered in the training set ("synthetic attacks"). The performance of the classifiers is measured using TAR (True Accept Rate) and corresponding FAR (False Accept Rate), where TAR is the proportion of legitimate users correctly classified and FAR is the proportion of non-users misclassified as legitimate users.

The usual evaluation follows a "zero-effort attacker model," where the attacker has no knowledge of their victims' behavior and spends no effort to bypass the system. However, a determined adversary may spend some effort to bypass the victim by studying their behavior or submitting samples that may be closer to those from the victim. The performance of attacks on these systems has been reported using the increase in EER (Equal Error Rate) (the operating point where the classifier's FAR is equal to its FRR (False Reject Rate) [Serwadda et al. 2016] or the bypass success rate or percentage of population compromised (the fraction of targeted users who were successfully attacked) [Khan et al. 2016]. In addition to capturing the success of attacks, mean attempts to bypass captures the expected number of failures before a successful bypass of the system.

Attacks on biometrics, where the attacker spends effort for a successful attack, can be mounted using different techniques. Zhao et al. [2020] propose a random input attack that exploits the fact that the acceptance region for biometric classifiers is usually larger than the true positive region due to unlabeled space. Poisoning attacks on behavioral biometrics are also possible, where the attacker exploits the behavior template update procedure [Lovisotto et al. 2020]. Two of the more common attacks are imitation and statistical attacks, which are discussed in the following.

2.1 Imitation Attacks

In imitation attacks, human attackers attempt to mimic the behavior of the target victim to defeat the authentication system. To mount the imitation attack, an attacker either needs to observe the victim [Hautamäki et al. 2015; Stang 2007] or have access to victims' behavioral data [Gafurov et al. 2007; Tey et al. 2013]. However, statistical attacks can be mounted without any knowledge of the victims' behavior, as they leverage overlaps in the behavior of the general population. Researchers have evaluated imitation attacks against the gait, voice, keystroke, and touch input-based biometrics [Gafurov et al. 2007; Khan et al. 2016; Panjwani and Prakash 2014; Tey et al. 2013].

Gafurov et al. [2007] used crowdsourcing techniques to evaluate the susceptibility of the gait biometric to imitation-based attacks. Using data from 90 participants, they showed that matching subjects against a set of physical characteristics could increase the EER from 0.12 to 0.22. Stang [2007] mounted an imitation-based attack on gait biometrics using a multimedia projector. In their experiment, 13 participants attempted to mimic 4 victims, each with 15 attempts of imitations to achieve an average bypass success rate of 42%.

Panjwani and Prakash [2014] mounted an imitation attack on the voice biometric using the MTurk platform and human imitators. They were able to find nine imitators, six through MTurk and three through human agents from a pool of 176 and 25, respectively. The nine candidates had a bypass success rate of 33%. Hautamäki et al. [2015] also targeted the voice biometric using professional mimickers. They used non-expert human listeners along with three different speaker verification systems and reported that the EER increased by almost 100%.

Tey et al. [2013] trained attackers to mimic victims' keystroke patterns on physical keyboards by practicing them while getting feedback in a user interface. They showed that with the partial knowledge of the keystroke patterns of the victims, 14 of their best attackers (out of 84 attackers) were able to bypass the system with a 99% success rate. An approach to train attackers using a user interface was also utilized by Khan et al. [2016] against touch input biometrics. They developed an application to mimic the touch input behavior and recruited 32 participants to achieve a bypass success rate of 86%. An augmented reality based approach was used to target keystroke biometrics against a virtual keyboard on smartphones [Khan et al. 2018, 2020]. The attackers trained by an augmented reality based guidance system were able to achieve a bypass success rate of 87% against virtual keyboards on smartphones [Khan et al. 2018]. A similar setup was used to bypass the touch input biometric with a bypass success rate of 99% [Khan et al. 2020].

Despite their high success rates, imitation attacks require the behavior of the victim, which may not always be possible.

2.2 Statistical Attacks

The key idea behind statistical attacks is that biometrics are not truly distinct and may have a considerable behavioral overlap across the general population. Statistical attacks infer general population statistics to attack the victim and have been demonstrated against the handwriting, keystroke, touch, and gait biometrics. Statistical attacks are demonstrated with datasets that do not include victims' data and only include data from the general population [Negi et al. 2018; Serwadda and Phoha 2013a; Serwadda et al. 2011].

Ballard et al. [2006] targeted handwriting biometrics. They collected 11,038 handwriting samples from 50 users using a digitized pen tablet. They used human forgers and a generative model for attacks. Their generative model takes the victim's style information (e.g., "cursive" vs. "block" writer), randomly chooses another user with the same style, and then uses statistics of this user to generate attack samples. If the attack is unsuccessful, another user is chosen. They showed that the nine best human forgers raised the EER from 5.5% for the zero-effort attack to 20.6%. The generative attack increased the EER to 27.4%. Unlike Ballard et al.'s choice of using a random person from a subset of users, like several recent approaches, we use statistical properties of the population (and not a target user) to generate attack samples. Furthermore, our approach efficiently navigates the search space instead of making random selection.

Keystroke behavior like other biometrics is not quite unique, and exploitable overlaps exist across the general population. Serwadda et al. [2011] attacked the keystroke biometric on personal computers using population statistics inferred from more than 3,000 users. Their attack ("MasterKey") models keystroke biometric using a Gaussian distribution, identifies the mean of the data, uses this mean as the first attack attempt, and then uses a step size based on the standard deviation to gradually move away from the mean for subsequent attempts. They found that with a single guess they can breach approximately 5% to 30% of the users. In their follow-up work [Serwadda and Phoha 2013a], they performed a statistical analysis of the keystroke biometric. This version of MasterKey increased the mean EERs of the three high performing keystroke classifiers between 28% and 84%. Due to its construction, MasterKey is not effective against victims who are far from the population mean, as it requires a large number of attack attempts to bypass them.

Negi et al. [2018] improved the performance of MasterKey by navigating the population characteristics more efficiently. They modified the K-means++ initialization algorithm (where the initial points are farther away from each other and have a higher probability of landing in different clusters) [Vassilvitskii and Arthur 2006] to efficiently find samples that match with the victim's behavior among the general population. They considered two variants of K-means++. The targeted version assumes that the attacker has many samples of the keystroke behavior for the victim's password mined from the general population using crowdsourcing. The indiscriminate version uses the probability distribution derived from the general population to generate samples. With the targeted variant, they were able to compromise the security of 40% to 70% of users within 10 attempts or less. The indiscriminate version was able to breach 30% to 50% of the users within 10 attempts.

Serwadda and Phoha [2013b] showed the vulnerability of the touch input biometric against input derived from general population statistics. After mining general population characteristics, they used a Lego robot to feed the input to a smartphone. They reported an increase in the classifier's EER between 339% to 1,000% compared to a zero-effort attack. In their follow-up work [Serwadda et al. 2016], they evaluated the effectiveness of their statistical attack on seven different touch input classifiers. They reported that the FAR increased from between 0.17 and 0.32 to over 0.70.

Zhu et al. [2020] showed that the gait biometric is also vulnerable to statistical attacks. They considered two attack scenarios. In the first scenario, they repurposed the K-means++ approach of Negi et al. [2018] for their attack, which does not require the knowledge of the victim's behaviour. They showed that against different gait classifiers, it was able to compromise 15% to 60% of the victims within five attempts. In the second scenario, with some knowledge of the victim's gait pattern, the attack was able to compromise 20% to 80% of the users within five attempts against different classifiers.

Stanciu et al. [2016] designed a statistical attack that uses a combination of feature weights and data binning to generate attack samples. Their attack bins each individual feature and generates attack samples based on the size of bins and feature weights as learned by a **Support Vector Machine (SVM)** classifier. For demonstrating the attack, they collected keystroke and sensory (accelerometer and gyroscope) data from 20 participants. The collected data was used to generate attack samples and attacked two classifiers against three scenarios: (1) only keystroke features, (2) only gyroscope and accelerometer features, and (3) a multimodal combination of keystroke and gyroscope and accelerometer features. The attack resulted in an EER between 31.5% and 50.6% for the keystroke feature only, 0.2% and 3.9% for sensory features only, and 0.2% and 14.9% for a combination of features. Therefore, they argued that the keystroke biometric should be complemented with sensory biometric to defend against statistical attacks. We do not compare with Stanciu et al.'s attack, as they report that their attacks fail for SVM and Naive Bayes classifiers. Unlike their defense that relies on sensory biometric, our detection mechanism is generic and works against different biometrics (see details in Section 5).

Acar et al. [2020] also used accelerometer and gyroscope sensors on a smartwatch to augment keystroke authentication. They collected keystroke and sensory data from 34 participants using an Android smartwatch. In addition to evaluating using the zero-effort attack model, they also mounted imitation and statistical attacks. For imitation attacks, participants watched typing videos of the victim and imitated their behavior. For statistical attacks, they used the same approach as Stanciu et al. [2016], with three bin sizes (5, 50, 500). They reported acceptance rates between 0.025 and 0.11 for the zero-effort attack, between 0.035 and 0.115 for the imitation attack, and between 0.02 and 0.078 for the statistical attack. They concluded that statistical and imitation attacks against their approach were only about as effective as a zero-effort attack.

Statistical attacks are a serious threat since they do not require the knowledge of the victim's behavior and are successful against a significant proportion of the population [Negi et al. 2018;

Serwadda and Phoha 2013a; Serwadda et al. 2011, 2016]. The existing approaches have three limitations. First, the degree of overlap is not quantified, which helps identify the most promising attack samples. Second, most existing approaches work well when most of the population is close to the population mean or centroid. If the population is distributed across multiple clusters, there may be multiple overlapping regions and the attacker needs to navigate the entire cluster space for victims who are not close to the population mean. By identifying overlapping regions within clusters, our approach maximizes the likelihood of attackers' success in as few attempts as possible (see Figure 1 and the discussion in Section 4). Third, these approaches suffer from multiple failed attempts for a considerable user population before a successful bypass. Finally, none of the existing works discuss how to detect these attacks and only show the limited efficacy of these attacks against multiple biometrics [Acar et al. 2020; Stanciu et al. 2016].

3 THREAT MODEL

We assume that the attacker targets a specific victim and aims to bypass the victim (i.e., the attacker is not mounting a spray attack). Similar to previous works, the goal of the adversary is to gain access in as few attempts as possible to not raise an alarm or get locked out [Acar et al. 2020; Khan et al. 2018; Negi et al. 2018]. The targeted behavior-based authentication system constructed using a machine learning classifier is employed to authenticate the user. The system may need to aggregate multiple samples for one binary decision. This system could be deployed as a web service ("user-to-remote service authentication") or on a device ("user-to-device authentication"). For user-to-remote service authentication, like other attack proposals [Serwadda et al. 2011; Tey et al. 2013], we assume that the adversary has access to the API to the biometric system, which is trained with the data of the target user and some random negative users. This API accepts a feature vector and provides a binary outcome (accept or reject). For user-to-device authentication, similar to other attack proposals [Khan et al. 2016, 2018; Serwadda and Phoha 2013b], we assume that the adversary needs to submit raw biometric samples to the system to receive the binary outcome. To submit data to the remote service, the adversary can use a bot, like the assumption by Tey et al. [2013].

For user-to-device authentication, the adversary needs physical access to the device often for a longer duration. The adversary can use different methods to submit attack data without rooting or installing anything on the device. For instance, to submit touch or keystroke inputs, the adversary can use a smartphone-to-smartphone augmented reality based setup [Khan et al. 2018, 2020]. For the voice biometric, adversaries can use methods proposed by Gao et al. [2018] to generate voice samples with certain characteristics. For the gait biometric, Zhu et al. [2020] proposed different ways an attacker can submit gait data without rooting the device including using a simple robotic body [Serwadda and Phoha 2013b], imitation based on human training, and using `SMAShed` [Mohamed et al. 2016] to directly manipulate motion sensors on an unrooted Android device via the Android debug bridge. Like previous works, we assume that if the device or the remote service is protected using a primary authentication mechanism (e.g., a PIN or password), the attacker has its knowledge (both username and the corresponding secret) and the knowledge of the features used by the model [Khan et al. 2016, 2018; Serwadda and Phoha 2013b; Serwadda et al. 2016; Tey et al. 2013].

The adversary does not have any behavioral samples from the victim and has no information about the distribution of the victim's behavioral data. The adversary uses population statistics derived from a pool of behavioral biometric data to find a feature vector (or corresponding raw data) using one or more techniques for which the system accepts the user. It should be noted that this pool of data *does not include the data of the victim* and may be collected from public sources including crowdsourcing platforms like Amazon MTurk. However, the general population data needs to be for the same scenario for the same biometric. For instance, if the biometric authentication scheme is for free-form swiping during the normal device use, then the general population data from the swiping behavior of a banking application only may not work as desired.

4 STATISTICAL ATTACKS USING HYPERVOLUMES

Previous approaches to statistical attacks have been designed for specific biometrics and do not quantify overlap and are ineffective against population samples that are farther from the mean or centroid (see Section 2). Since overlap is fundamental to statistical attacks, we use hypervolumes to model and exploit overlap. A hypervolume is a region defined by three or more dimensions in the n-dimensional space and can be considered as an n-dimensional geometric shape [Blonder 2018]. Hypervolumes have been widely used in the field of ecology and evolution since the proposal of Hutchinson [1957]. We propose to use hypervolumes to quantify overlaps in n dimensions to identify regions where overlapping user biometric behavior exists. Once we quantify overlaps, we use this information to generate attack samples.

Despite the intuitive nature of the concept, determining how to estimate the shape and related operations on hypervolumes has proven to be difficult [Blonder et al. 2018]. Multiple methods for calculating hypervolumes are available—each with its own underlying assumptions. Existing approaches for calculating hypervolumes use dynamic range boxes [Junker et al. 2016], convex hulls [Villéger et al. 2008], multidimensional ellipses [Swanson et al. 2015], Gaussian kernel density estimation [Blonder et al. 2018], and one-class SVM [Blonder and Harris 2019]. We choose the dynamic range boxes technique proposed by Junker et al. [2016], as it provides several desirable properties, including (1) it is independent of the data distribution, (2) it is robust against outliers, (3) it is applicable for data of different dimensions and produces dependable results independent of the data dimensionality, and (4) it provides information about individual dimensions. More details are provided next.

4.1 Hypervolumes Using Dynamic Range Boxes

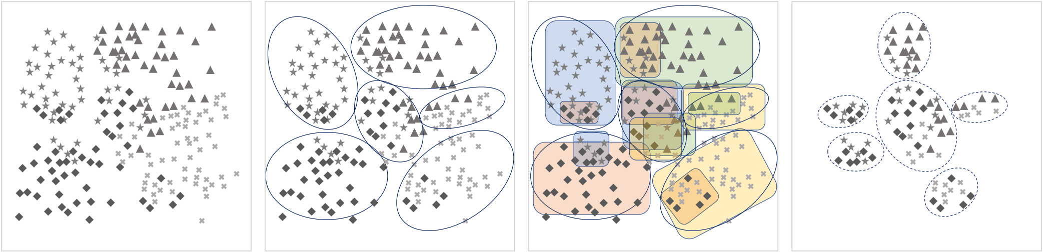
In this section, we describe the dynamic range box approach to hypervolumes in the context of behavior biometrics using the notation and methods from Junker et al. [2016].

Let $A = (a_i)_{i=1}^n$ and $B = (b_i)_{i=1}^n$ be two n-dimensional matrices containing biometric data captured as an array of features for two users A and B. These matrices are standardized to the n-dimensional unit box $[0,1]^n$, by calculating the minimum, $min(min(a_i), min(b_i))$, and maximum, $max(max(a_i), max(b_i))$, for each dimension and using these numbers to translate and then scale all points in each dimension separately. The corresponding standardized matrices are $\hat{A} = (\hat{a}_i)_{i=1}^n$ and $\hat{B} = (\hat{b}_i)_{i=1}^n$. If more than two users are involved, the minimum and maximum are taken over all users.

For simplifying the notation, let $X = \hat{A}$, and by definition, α-quantile of $X = (x)_{i=1}^n$ is a feature vector represented as $F_X^-(\alpha) = (F_{X_i}^-(\alpha))_{i=1}^n$. For each dimension $i = 1, \ldots, n$ and $\alpha \epsilon [0,1]$, the α-range interval $I_i(\alpha)$ is given by

$$I_i(\alpha) := \left[F_{x_i}^-\left(\frac{1-(1-\alpha)^{\frac{1}{n}}}{2} \right), F_{x_i}^-\left(1 - \frac{1-(1-\alpha)^{\frac{1}{n}}}{2} \right) \right]. \tag{1}$$

The n-dimensional range box is then defined as the Cartesian product of the intervals $I_1(\alpha), \ldots, I_n(\alpha)$, denoted as $R_n^A(\alpha) := X_{i=1}^d I_i(\alpha)$. The range box $R_n^B(\alpha)$ is calculated similarly. This choice of interval in each dimension ensures that (under the independence assumption) the range boxes cover $100(1-\alpha)\%$ of the data. It is well established that the empirical quantile function, $F_X^-(\alpha)$, converges weakly to the true quantile function [Van der Vaart 2007]. Using dominated



(a) Sample data distribution (b) Data clusters (ovals) (c) Hypervolumes (rectangles) (d) Overlapping regions

Fig. 1. A simple 2D representation of the proposed approach. (a) Data distribution of four users in two dimensions. (b) Data is clustered to reduce the overlap calculations (shown using ovals). (c) Hypervolumes are calculated for each user within each cluster (shown using rounded rectangles, where same colors represent the same user). (d) Overlapping regions are identified using intersecting hypervolumes (shown using dashed ovals).

convergence [Klenke 2007], we directly get consistency of users' behavior overlap—that is, empirical overlap converges to true overlap with probability 1.

Let V denote the n-dimensional volume (i.e., the product of the side length of the boxes). We are interested in the average portion of the biometric behavior of user A that is covered by the behavior of user B, and the average portion of the biometric behavior of user B covered by user A. These quantities are defined as

$$overlap(A, B) := \int_0^1 \frac{V(R_n^A(\alpha) \cap R_n^B(\alpha))}{V(R_n^B(\alpha))} d\alpha, \tag{2}$$

$$overlap(B, A) := \int_0^1 \frac{V(R_n^B(\alpha) \cap R_n^A(\alpha))}{V(R_n^A(\alpha))} d\alpha. \tag{3}$$

The volume of user A is defined as the average portion of A in the uniform distribution U on $[0, 1]^n$, which is given by

$$vol(A) := overlap(A, U) = \int_0^1 min \left\{ \frac{V(R_n^A(\alpha))}{1 - \alpha}, 1 \right\} d\alpha. \tag{4}$$

Therefore, the minimum in the integrand makes sure that the quotient cannot exceed 1. Setting $\alpha_i = \frac{i-1}{m}$ for every $i = 1, \ldots, m$, these integrals can both be well approximated using the trapezoidal rule.

Interested readers are referred to the original paper [Junker et al. 2016] for the further details and design goals. We used the R package DynRb [Schreyer et al. 2018] related to the original paper [Junker et al. 2016] for calculating hypervolumes in this work.

4.2 Attack Intuition

Figure 1 provides a simplified version of our proposed approach using four users with two features for each user's behavior. Figure 1(a) shows the data distribution for the four users. There are overlaps in behavioral regions in our example—as shown to be expected by previous works on biometrics [Negi et al. 2018; Serwadda and Phoha 2013b; Serwadda et al. 2011]. Users' behavior depending on the biometric used can exist in multiple local clusters throughout the population space. To calculate hypervolumes in these local clusters, we first create clusters of the population data (see Figure 1(b)). Next, hypervolumes are calculated for each user within each cluster (shown using rounded rectangles in Figure 1(c)). Calculating hypervolumes for each user separately for each cluster enables us to identify the regions in the n-dimensional space where the general user behavior is most likely to exist. Finally, we calculate the intersections of hypervolumes to determine the overlapping regions (i.e., where hypervolumes intersect each other). Figure 1(d) shows samples in these regions. Attack samples are generated based on the properties of these regions.

Algorithm 1 provides the detailed construction of our attack. It takes a dataset, *which does not contain the target victim's data in it*. We first cluster data using the K-means clustering algorithm with K-means++ initialization [Vassilvitskii and Arthur 2006] (lines 2 and 3). The K-means algorithm requires the number of clusters. For selecting the number of clusters, we evaluated two methods for minimizing the within cluster sum of squared distances from the centroid—silhouette analysis [Rousseeuw 1987] and elbow methods [Satopaa et al. 2011]. We compared the performance of two clustering techniques by comparing the percentage of population compromised in the first attempt. For the majority of attacks, the percentage of population compromised was higher when the data subset was partitioned using optimal number of clusters found by the elbow method. Therefore, we chose the elbow method. Next, we calculate the hypervolume for each user, overlaps among hypervolumes in each cluster, and cluster score, S, for each cluster (lines 4–6; for details, refer to Algorithm 2).

Algorithm 2 calculates hypervolumes and overlaps between them using `DynRb` [Schreyer et al. 2018] (Algorithm 2: line 5). Overlap information is used to extract samples from the overlapping regions (Algorithm 2: line 10). Finally, each cluster is scored based on three characteristics: (1) the mean overlap within a cluster, which identifies clusters with most overlapping samples; (2) the number of unique users in a cluster; and (3) the number of samples in a cluster (Algorithm 2: line 12). This scoring will be used for ranking the clusters. The ranking provides a sizable cluster with a high overlap of a larger number of users. A weighted average of the three criteria is used for our ranking. Weights used in our experiments are (1) 0.9 for the mean overlap, (2) 0.05 for the number of unique users, and (3) 0.05 for the number of unique samples in a cluster. These weights were selected using a randomized grid search. The search vector for each weight was generated by splitting the range of 0 to 1 into 20 equally spaced points. The metric used to evaluate the performance of weights was the percentage of population compromised. Next, we rank the clusters to identify the order in which attack samples from each cluster will be extracted (Algorithm 1: line 7). Ranking is done by sorting the cluster scores in descending order.

We then generate the first few attack samples using the centroids of the overlap clusters, $\vec{\mu}$, in descending order (highest-ranked clusters are attempted first) (lines 8–14). If the centroids of overlap clusters are exhausted, we adopt the following strategy. According to the cluster ranking in each cluster, the distance between overlapping samples, O, and the last tried samples is calculated (lines 17, 18). A new attack sample is selected until a successful sample is found using a strategy like the modified K-means++ approach of Negi et al. [2018] (lines 19–21). If a sample is unsuccessful, it is removed from the cluster's overlapping samples (line 24). Algorithm 1 terminates when all samples in the cluster's overlapping samples are exhausted (line 25).

We note that our attack technique requires the calculation of hypervolumes and overlaps for each pair of users, which is computationally expensive and depends on the size of the feature space. For different datasets used in this work, these calculations may take up to several hours on a personal computer. However, this is a one-time cost, and the performance of the hypervolume attack to crack users within a small number of attempts (see Section 6) justifies this computational cost.

5 EXPERIMENTAL SETUP FOR ATTACKS

In this section, we provide the details of our attack setup including biometrics, datasets, and machine learning classifiers. After establishing an evaluation baseline, we discuss the other

ALGORITHM 1: Mount Hypervolume Attack

```
Data: $\chi$: Attacker's Biometric data
Result: True if attack is successful else False
Function MOUNT_HYPERVOLUME_ATTACK($\chi$):
    $k \leftarrow elbow\_method(\chi)$                ▷ $k$ is number of clusters
    $C \leftarrow k\text{-}means(\chi, k)$
    for $i \leftarrow 1, \ldots k$ do
        ▷ Provides cluster scores, overlap region samples of each cluster, and mean of overlap region
        samples as described in Algorithm 2
        $S_i, O_i, \vec{\mu_i} \leftarrow GET\_CLUSTER\_SCORE(C_i)$
    $sort(S, O, \vec{\mu})$        ▷ Sort by descending cluster score
        ▷ Try mean of overlap region samples:
    for $t \leftarrow 1, \ldots k$ do
        $\mathcal{T} \leftarrow \vec{\mu_t}$        ▷ $\mathcal{T}$ is attack sample; $\mathcal{V}$is target victim
        if $\mathcal{T}$ is accepted for $\mathcal{V}$ then
            return True
        else
            $\vec{x_t} \leftarrow \vec{\mu_t}$        ▷ Setting last tried point to overlap region centroids
    while samples_exist do
        for $r \leftarrow 1, \ldots k$ do
            ▷ Compute distances from last tried point to each sample
            $D \leftarrow$ COMPUTE_DISTANCES$(O_r, \vec{x_r})$
            $\mathcal{T} \leftarrow \vec{x}$ with probability $\left(\frac{D(\vec{x_r})^2}{\sum_{x' \epsilon O_r} D(\vec{x'_r})^2}\right)$
            if $\mathcal{T}$ is accepted for $\mathcal{V}$ then
                return True
            else
                $\vec{x_r} \leftarrow \vec{x}$
                remove sample $\vec{x}$ from $O_r$
    return False
```

statistical attack approaches that we compare against, the metrics used for the comparison, and our experimental setup.

5.1 Targeted Biometrics

We target five popular biometrics: touch input, keystroke dynamics, mouse movement, gait, and voice. Our choice of biometrics is diverse as it includes both physiological (voice) and behavioral biometrics. The behavioral biometrics capture behavior using users' interaction data as well as gait patterns using onboard motion sensors. We do not target face and fingerprint biometrics, as these biometrics are more unique than behavioral or voice biometrics [Jain et al. 2004]. A brief description of each scheme, including its features and reported evaluation results, is provided next.

5.1.1 Touch Input. Touch input schemes use finger movement patterns during normal device usage to build a profile of the user. We employ Touchalytics [Frank et al. 2013], which extracts 31 features from the raw touch data of each swipe. These features capture the behavior of the

ALGORITHM 2: Get Cluster Score

```
Data: $C$: One cluster of biometric dataset samples
Result: Cluster score, overlapping region samples, and overlapping regions mean
Function GET_CLUSTER_SCORE($C$):
    ▹ Gather control values:
    $U \leftarrow$ count_unique_users_in_cluster($C$)
    $m \leftarrow compute\_number\_of\_samples(C)$
    $\vec{OS} \leftarrow$ compute_hyp_overlaps_each_user_pair($C$)
    $w_u \leftarrow$ get_weight_for_number_of_users($U$)
    $w_m \leftarrow$ get_weight_for_number_of_samples($m$)

    $\mu_{OS} \leftarrow mean(\vec{OS})$
    $w_{OS} \leftarrow$ get_weight_for_overlap_region_mean($\mu_{OS}$)
    $O \leftarrow get\_overlap\_region\_samples(C, \vec{OS})$
    $\vec{\mu_o} \leftarrow mean(O)$
    $s = \mu_{OS} * w_{OS} + U * w_u + m * w_m$
    return $s, O, \vec{\mu_o}$
```

user using the swipe direction, swipe location, duration and length of the swipe, velocity and acceleration of the swipe, curvature of the swipe, touch area and touch pressure of the swipe, and the orientation of the finger and the device. An evaluation of Touchalytics on a dataset of 41 users shows that when using either an SVM or a **K-Nearest Neighbors (KNN)** classifier for authentication on this dataset that these provide an EER of 4% for a window of eight swipes.

5.1.2 Keystroke Dynamics. Keystroke dynamics approaches have been proposed for physical keyboards on computers and smartphones [Clarke and Furnell 2007; Killourhy and Maxion 2009]. Buschek et al. [2015] also proposed a keystroke dynamics scheme for virtual keyboards on smartphones; however, several features for virtual keyboards overlap those of Touchalytics. Therefore, we choose the scheme of Killourhy and Maxion [2009] for physical keyboards of a personal computer. They capture the keystroke behavior of each key using two features: key hold interval and inter-stroke interval. These features have been widely used by other schemes and previous statistical attacks on keystroke dynamics [Serwadda and Phoha 2013a; Serwadda et al. 2011]. On a dataset of 50 users entering a password on a laptop, they show an EER of 10% using SVM.

5.1.3 Mouse Movement. For mouse movement behavior, we choose the scheme proposed by Zheng et al. [2011]. They capture mouse movement behavior using 21 features including the stroke curvature, speed, click duration, and acceleration [Zheng et al. 2011]. Their evaluation of the mouse biometric on two datasets using SVM shows an EER of 1.3% using 20 clicks.

5.1.4 Gait. Data from accelerometer and gyroscope sensors on smartphones has been used to characterize gait behavior of users. Thang et al. [2012] proposed a gait pattern based classifier that captures gait patterns using the first 40 Fast Fourier Transform coefficients calculated on eight consecutive gait cycles. They used an SVM classifier on data from 11 users to achieve a classification accuracy of 92%.

5.1.5 Voice. Speaker verification in an unconstrained environment is a challenging problem [Nagrani et al. 2020]. [Nagrani et al. 2020] created a dataset, VoxCeleb, using the recordings of celebrities. Over samples of 1,251 speakers, they reported an average EER of 7.8% using a convolutional neural network. Chung et al. [2018] proposed a method based on a residual neural network and evaluated it on the VoxCeleb2 dataset (with 6,112 speakers) to report an average EER of 3.95%.

5.2 Datasets

For our analysis, we used publicly available datasets. A brief description of the datasets follows.

5.2.1 Touch Input. Sitová et al. [2015] collected raw touch data from 100 participants across eight sessions in a lab environment. During each session, the participants were randomly assigned a reading, writing, or map navigation task. Their touch interaction data was collected on an Android device and made publicly available. Since there was insufficient data from one user, we used data from the remaining 99 users. In our data subset, there are 1,393 samples per user on average.

5.2.2 Keystroke Dynamics. We used the keystroke dataset collected by Killourhy and Maxion [2009]. This dataset has key hold and inter-stroke intervals from 51 participants, where they typed a strong password, ".tie5Roanl." The data was collected on a Windows XP laptop across eight sessions, where participants entered the password 50 times during each session. In total, each participant typed the password 400 times.

5.2.3 Mouse Movement. For the mouse biometric, we used data collected by Eberz et al. [2018]. They recorded data on a Windows machine through the `PyHook` Python module, which uses the Windows hooking API. Data was collected from 59 participants, where each participant played a mole clicking game 250 times.

5.2.4 Gait. For the gait biometric, we used data collected by Zou et al. [2020]. This data was collected from 118 participants and contains accelerometer and gyroscope readings on a smartphone. This data has an average of 155 samples per user.

5.2.5 Voice. The extracted voice biometric features of VoxCeleb and VoxCeleb2 are publicly available. We used the test portion of the VoxCeleb2 dataset [Chung et al. 2018], which contains 118 users with an average of 406 utterances per user.

5.3 Classifiers, Parameter Selection, and Zero-Effort Performance Baseline

We evaluate attacks against SVM, KNN, **Random Forest (RF)**, and **Deep Learning (DL)** classifiers. These classifiers have been used for different biometrics evaluated in this work [Negi et al. 2018; Serwadda and Phoha 2013b; Serwadda et al. 2016; Zhao et al. 2020]. The original proposal for the voice biometric [Nagrani et al. 2020] used a DL classifier only. However, we use both traditional and DL classifiers across all biometrics for a systematic comparison. For DL models, we used the `TensorFlow` [Abadi et al. 2016] library, whereas the remaining classifiers were evaluated using the Python `sklearn` [Buitinck et al. 2013] library.

For evaluations, we split each of the datasets into two halves with an approximately equal number of users in each. One half is used for inference to generate attack samples ("inference subset") and the other half for evaluating these attacks ("evaluation subset"). This simulates a real attack scenario, where the attacker has no prior knowledge of a victim's samples. We also cross validate by switching the role of the two halves and report the average results.

In the evaluation subset, each users' data (positive samples) is combined with other users' data (negative samples) in the same subset to get a balanced dataset for each user. This balanced dataset is then split into training and test sets (80:20 split). We ensure that the training data temporally precedes test data to avoid any temporal bias. A different classifier for each user is trained using random grid search of the parameters with 10-fold cross validation. This step ensures that the classifier is at its best operating point against the attack. The base line for the performance is established by testing the trained classifier on the test data for that user. We see that our grid search converges to a subset of points for most of the users. For instance, for SVM, on average,

Table 1. Details of the DL Model

Layer	Layer Type	Feature Map
Dense 1	Fully connected	1×768
Normalization 1	Normalization	1×768
Drop 1	Dropout	1×768
Reshape 1	Reshaping	$16 \times 16 \times 3$
Conv 1	Convolution	$14 \times 14 \times 32$
Pool 1	Max pooling	$7 \times 7 \times 32$
Drop 2	Dropout	$7 \times 7 \times 32$
Flat 1	Flatten	1×1568
Dense 2	Fully connected	1×512
Out	Fully connected	1×1

the C parameter converges to eight values for 90% of the users. More details about the parameters of each classifier are provided in the following. If a value of a parameter is not reported, then its default value for `sklearn` was used [Buitinck et al. 2013].

Table 2 shows the performance of each classifier using the AUC (Area Under the ROC Curve) and EER. For a better comparison, we report results for individual samples without combining multiple scores using a majority decision. The table shows that most classifiers perform similar to the original papers. Performance evaluation for attacks is discussed in Section 5.5.

SVM. We used the radial-basis kernel function. For the regularization parameter C, we performed a random grid search with 10-fold cross validation. The range of C was set between 10^{-3} and 10^4 and 100 evenly spaced points were used in the range on a log scale.

KNN. For the three KNN parameters, we used random grid search with 10-fold cross validation. These parameters and the ranges explored are (1) the number of neighbors between 1 and 50, (2) the leaf size used for computing nearest neighbors if a tree algorithm is selected by `sklearn` between 1 and 70, and (3) the power parameter for the Minkowski metric as $1, 2$.

RF. The six RF parameters and corresponding ranges that were used in the random grid search with 10-fold cross validation are (1) the number of trees in the forest between 200 and 2,000; (2) the maximum depth of the tree between 10 and 110; (3) the minimum number of samples required to split an internal node as $2, 5, 10$; (4) the minimum number of samples required to be at a leaf node as $1, 2, 4$; (5) the number of features to consider when looking for the best split as $auto$, $sqrt$; and (6) whether bootstrap samples are used when building trees (bootstrap) as $True$, $False$.

DL. Details of the model that we used are provided in Table 1. We used early stopping and dropout layers to regularize the model and the Adam optimizer for training the neural network.

5.4 Comparison with Other Attack Approaches

To compare the efficacy of our attack, we implemented the following three statistical attack approaches. We explain these attacks using Figure 2, which provides a simplified 2D example for the attacks considered in this work. Figure 2 shows the data from the general population that the attacker possesses as black circles. The data for Victim A is shown using indigo left triangles, and the data for Victim B is shown using blue down triangles. Victim A's data distribution is very close to the mean of the general population data. The data distribution for Victim B is away from the mean of the attackers' data but still lies within the overlapping regions of the general population data. Finally, the attack samples are shown using shapes and colors according to the order in which they are generated (see the legend for Figure 2 for details). The attack samples are also numbered in the order in which they are generated.

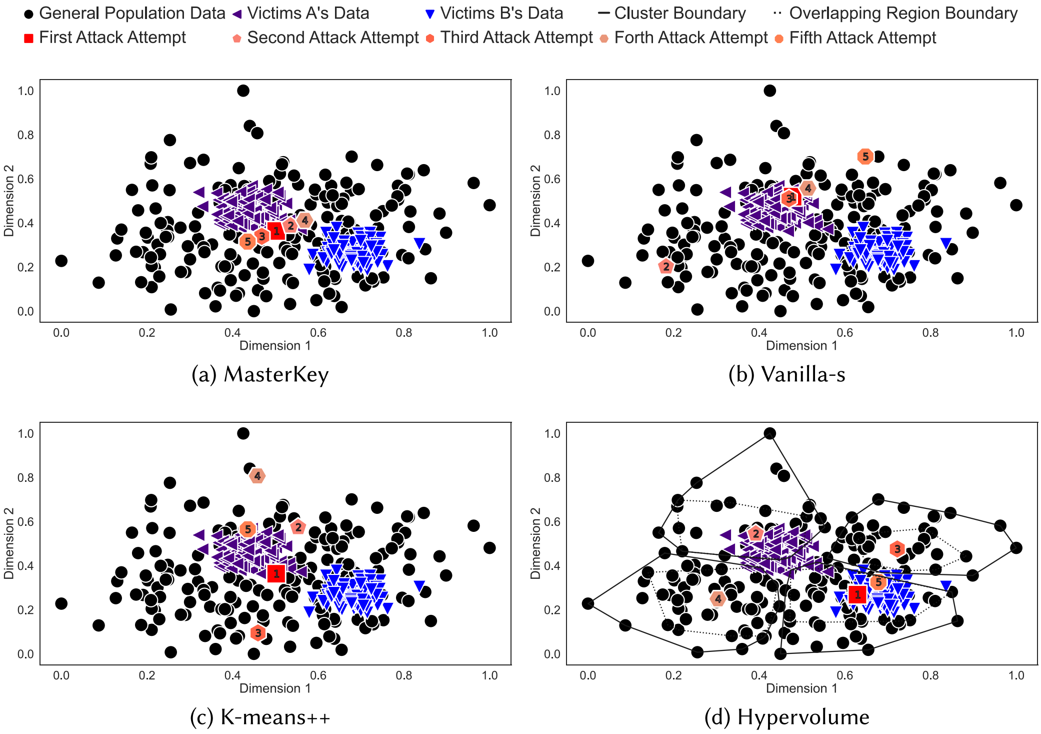
Table 2. Baseline Performance of Classifiers on Biometrics

Biometric	Metric	SVM		KNN		RF		DL	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
Touch	AUC	0.83	0.07	0.76	0.08	0.86	0.07	0.84	0.08
	EER	0.23	0.07	0.29	0.07	0.22	0.07	0.23	0.08
Keystroke	AUC	0.97	0.08	0.93	0.08	0.99	0.04	0.97	0.06
	EER	0.07	0.1	0.11	0.10	0.03	0.06	0.06	0.07
Mouse	AUC	0.77	0.10	0.72	0.08	0.83	0.08	0.74	0.10
	EER	0.29	0.10	0.34	0.07	0.24	0.08	0.32	0.09
Gait	AUC	0.94	0.07	0.90	0.08	0.97	0.05	0.92	0.09
	EER	0.11	0.09	0.15	0.09	0.08	0.07	0.13	0.09
Voice	AUC	0.99	0.01	0.95	0.05	0.99	0.01	0.99	0.02
	EER	0.03	0.03	0.08	0.08	0.04	0.03	0.04	0.04

5.4.1 MasterKey. Serwadda and Phoha [2013a] and Serwadda et al. [2011] proposed MasterKey, which generates attack samples by modeling the data of general population as a Gaussian distribution. Although MasterKey was proposed against keystroke dynamics, we evaluate it against all biometrics considered in this work. MasterKey uses the mean as the first attempt, shown as a red square in Figure 2(a). The first attempt will most likely compromise Victim A (indigo left triangle), due to the proximity of Victim A's samples to the mean of the population. If the first sample is not successful, then MasterKey navigates the attack space in steps based on a multiple of standard deviation away from the mean. This scenario is shown for Victim B (blue down triangle) in Figure 2(a), where MasterKey fails to compromise Victim B after five attempts. Unlike MasterKey, Hypervolume attack quantifies and prioritizes overlapping regions in the attack space.

5.4.2 Vanilla Statistical Attack ("Vanilla-s"). Serwadda and Phoha [2013b]; Serwadda et al. [2016] used population statistics estimated from large datasets to attack the keystroke and touch biometrics. Their work shows that there is significant overlap in the user behavior for keystroke and touch biometrics, which is used to estimate parameters for a Gaussian distribution. The learned distribution is used to randomly generate adversarial samples. To reliably estimate the parameters of a Gaussian distribution for adversarial sample generation, we need a large dataset. Due to the limited sample size of open source datasets, we use bootstrapping to estimate population statistics [Singh and Xie 2008]. The data from the inference subset is bootstrapped 30,000 times to estimate mean and standard deviation of features. Attack samples are then generated randomly using a Gaussian distribution with estimated mean and standard deviation. Figure 2(b) shows that the five attack samples for Vanilla-s are mostly near the mean of the general population. Therefore, Victim A with behavior similar to the general population mean is compromised in the first few attempts. Due to this attack's design, it is not able to transverse the attack space efficiently and it is able to mostly target victims with behaviors close to the mean of the general population. Since the data of Victim B (blue down triangle) is away from the mean of the general population, Vanilla-s failed to compromise it.

5.4.3 K-means++. Negi et al. [2018] proposed an attack that addresses the shortcomings of MasterKey and Vanilla-s of not navigating the attack space efficiently by repurposing the K-means++ initialization algorithm [Vassilvitskii and Arthur 2006]. The original K-means++ algorithm is meant to improve K-means clustering by choosing the initial cluster centroids that are not close to each other (to avoid local optima in the K-means clustering algorithm). The



(a) MasterKey

(b) Vanilla-s

(c) K-means++

(d) Hypervolume

Fig. 2. A simple 2D representation of the attack approaches. (a) MasterKey generates the first sample at the mean and generates remaining samples by moving in fixed steps away from the mean based on the standard deviation. (b) Vanilla-s generates the samples randomly from a Gaussian distribution. (c) K-means++ generates the first sample at the mean and generates remaining samples by selecting a point away from the current sample. (d) Hypervolume clusters the space and ranks the clusters, then it extracts overlapping regions from each of the clusters. The centroid of the highest rated cluster's overlapping region is used as the first attempt (for details, refer to Algorithm 1).

repurposed K-means++ algorithm of Negi et al. [2018] is provided as Algorithm 3. For their K-means++ attack, they use the mean of the general population for their first attempt (Algorithm 3: line 2). Figure 2(c) shows that Victim A (indigo left triangle) is most likely compromised in the first attempt, as it is closer to the mean of the general population. If the first attempt fails, like in the case of Victim B (blue down triangle), they use the K-means++ initialization algorithm to select the next attack sample (Algorithm 3: lines 5–9). Figure 2(c) shows that these samples are far from the previous samples, thus navigating the search space more efficiently. But since the search space is not reduced, the K-means++ attack is unable to compromise Victim B in the first five attempts. It should be noted that despite using the K-means++ initialization algorithm, K-means++ attack does not divide the search space into clusters. They evaluated their attack against the keystroke and touch biometrics and showed that it performs better than MasterKey.

Unlike existing approaches, our attack reduces the attack search spaces by extracting and prioritizing overlapping regions. Figure 2(d) shows the overlapping regions (dotted black lines) within the identified clusters (solid black lines) and how Hypervolume prioritizes regions where population clusters with a higher degree of overlap exist. Our toy example shows that due to its unique approach, Hypervolume attack is able to compromise both victims in just two attempts.

ALGORITHM 3: Adversarial Targeted K-means++ [Negi et al. 2018]

```
Initialize
    $Try_1 \leftarrow$ the mean of the collected adversarial dataset $\mathbb{X}$
    $Auth \leftarrow False$
    $i \leftarrow 2$
while $!Auth$ do
    $D(x) \leftarrow$ distance from nearest Try chosen so far to point $x$ ($\forall x \in \mathbb{X}$)
    $Try_i \leftarrow x \in \mathbb{X}$ with probability $\frac{D(x)^2}{\sum_{x' \in \mathbb{X}} D(x')^2}$
    $Auth \leftarrow True$ if $Try_i$ passes the authentication
    $i++$
```

In Section 6, we show that this approach of Hypervolume is more successful than other attack approaches against the biometrics evaluated in this work.

5.5 Evaluation Metrics and Setup

To measure the efficacy of the attacks, we choose two metrics used by previous works. First, we use the metric “Percentage of Compromised Population” to measure the percentage of the proportion of compromised users in each population against an attack. This metric is more meaningful when an attack’s success is complemented with its speed. To this end, we use “Attempts to Bypass” to capture the number of attempts required to compromise vulnerable users. A good attack should have a high proportion of compromised population coupled with fewer attempts to bypass. We used the same classifier parameters and training and test sets as provided in Section 5.3.

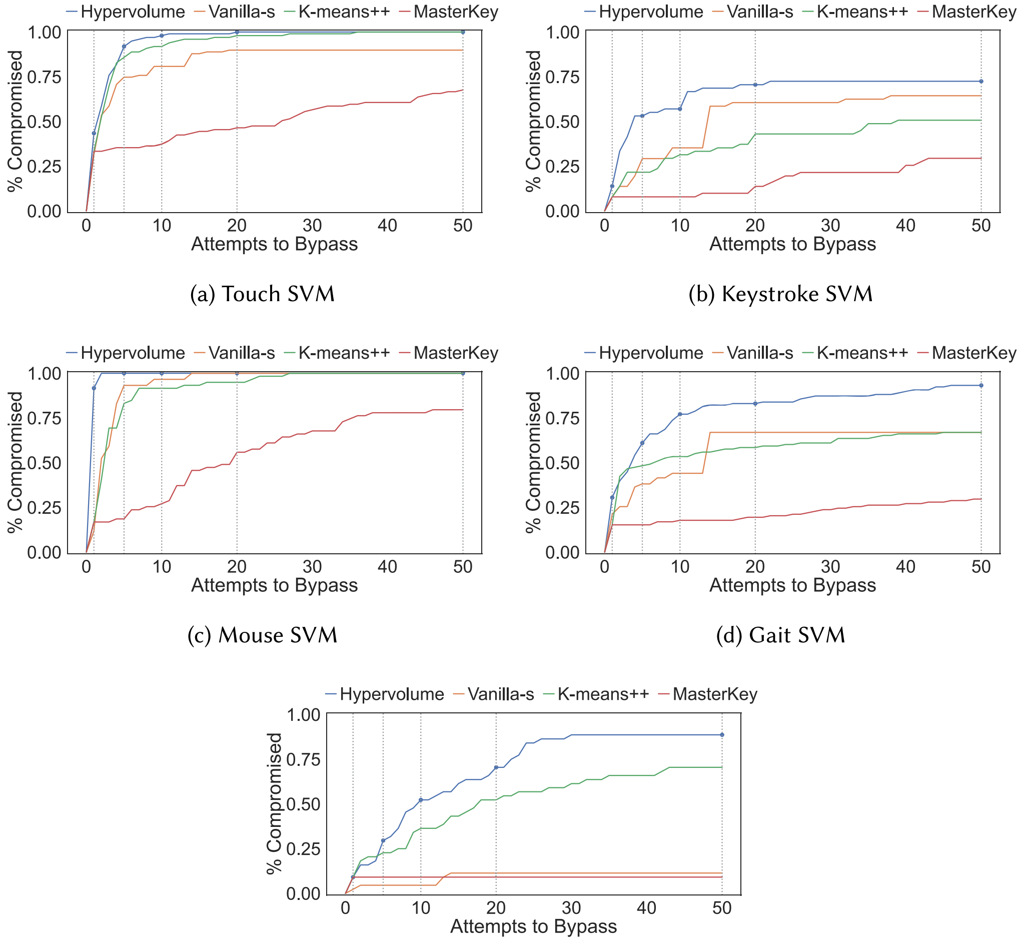
6 ATTACK EVALUATION

6.1 Attack Results

The performance of the attacks against the five biometrics considered in this work are shown using *Attack Curves* in Figures 3 through 6. Attacks that bypass the biometric in the first few attempts are better since they are difficult to detect [Acar et al. 2020; Khan et al. 2018; Negi et al. 2018]. For SVM, Figure 3(a), (b), (c), (d), and (e) show that for the first attempt, the hypervolume-based attack (“Hypervolume”) compromises 43%, 14%, 92%, 31%, and 9% of the population against the touch, keystroke, mouse, gait, and voice biometrics, respectively. For touch, keystroke, mouse, and voice, the second-best approach is a tie between K-means++ and MasterKey. This is because both attacks use mean of the inference data as first attempt. They compromised 33%, 8%, 16%, and 9%, respectively. For gait the second-best attack is Vanilla-s, which compromises 21% of the population.

These results indicate that choosing the population mean as the first attempt is not always the best approach, as Hypervolume is, on average, 21 percentage points more successful (37.8% vs. 16.4% average population compromised) than the second-best attack.

Hypervolume after the fifth attempt against SVM compromises 92%, 52%, 100%, 61%, and 30% of the population against the touch, keystroke, mouse, gait, and voice biometrics, respectively. K-means++ is the second most successful attack except for the mouse and keystroke biometrics where Vanilla-s outperforms it. In comparison, the proportion of population compromised for the second-best attacks for the fifth attempt are 86%, 29%, 93%, 48%, and 23% against the touch, keystroke, mouse, gait, and voice biometrics, respectively. For the 10th attempt, Hypervolume increases the compromised proportion of the population for the keystroke, gait, and voice biometrics to 57%, 77%, and 52%, respectively, whereas the second best are at 35%, 53%, and 36%, respectively. After 20 attempts, the proportion of compromised population for the touch and mouse biometrics by



(a) Touch SVM

(b) Keystroke SVM

(c) Mouse SVM

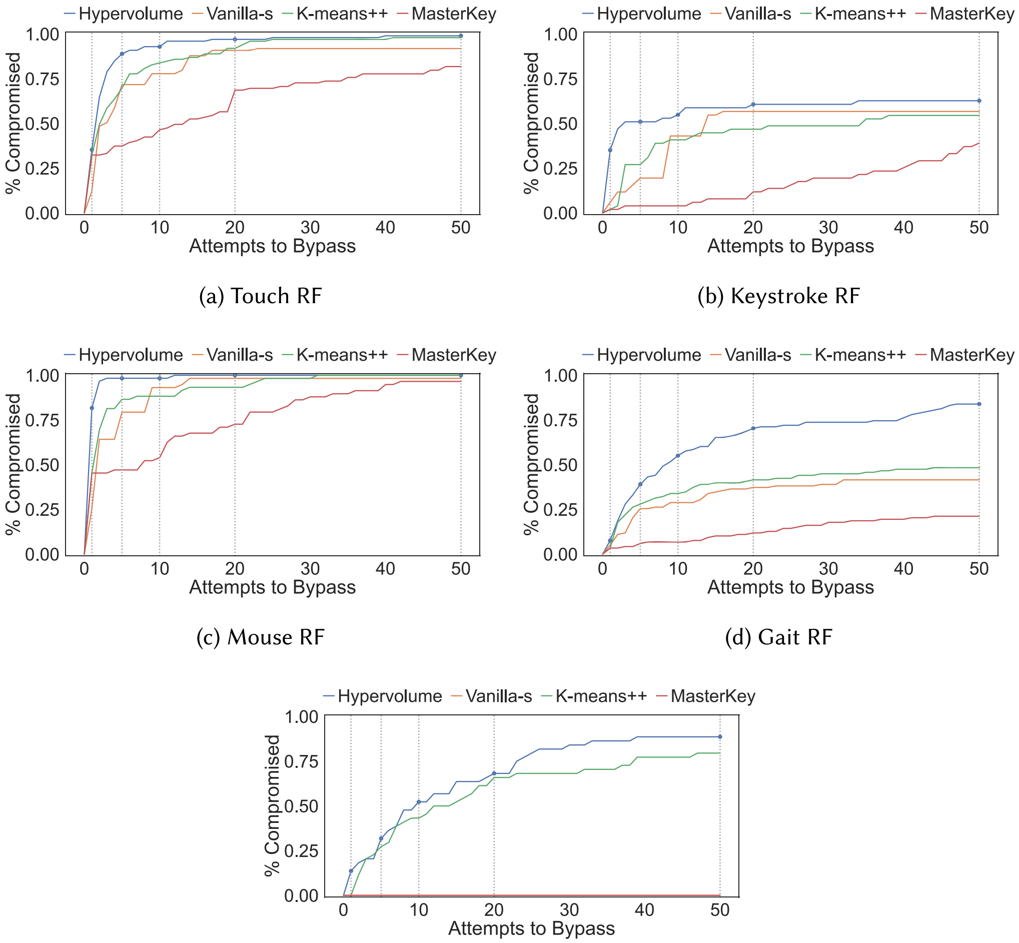
(d) Gait SVM

(e) Voice SVM

Fig. 3. Performance of attacks against five biometrics trained using SVM classifiers.

Hypervolume is 100%, and Vanilla-s and K-means++ compromise more than 90% of the population. MasterKey is only able to compromise 46% and 56%, respectively. After 50 attempts, we note that Hypervolume compromises 8%, 26%, and 19% more victims than the second-best attack for the keystroke, gait, and voice biometrics.

We observe quantitatively similar results for the other classifiers. For the RF classifier (see Figure 4, Hypervolume outperforms the other attacks against touch, keystroke, mouse, and gait biometrics in the first attempt by compromising between 3% and 37% more victims than the second best. For the voice biometric, it compromised 14% of the population in the first attempt, where all other attacks failed. We see similar attack performances for Hypervolume for the remaining attempts. For the KNN classifier (see Figure 5), we see an improved performance of the other attacks. However, Hypervolume provides better performance against all biometrics except the voice biometric (see Figure 5(e)). For the voice biometric, for the first 20 attempts, on average, K-means++ compromises 20% more victims than Hypervolume. However, after 20 attempts, the performance of Hypervolume improved, and after 50 attempts, it was able to compromise 5% more victims than K-means++.



(a) Touch RF

(b) Keystroke RF

(c) Mouse RF

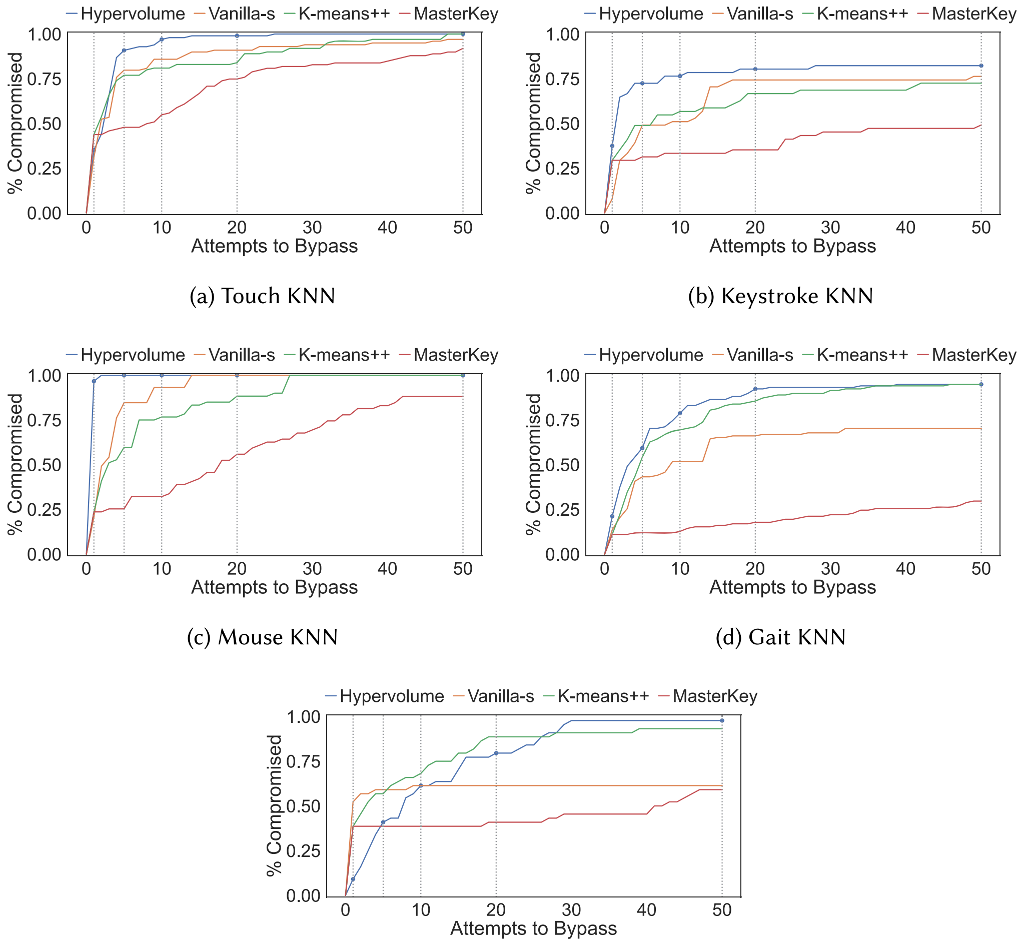
(d) Gait RF

(e) Voice RF

Fig. 4. Performance of attacks against five biometrics trained using RF classifiers.

For the DL classifier (see Figure 6), the first attempt of Hypervolume is significantly more successful compared to the other attacks across all biometrics and compromises 36%, 20%, 75%, 25%, and 16% of the victims for touch, keystroke, mouse, gait, and voice biometrics, respectively. The second-best attacks against touch and gait biometrics were K-means++ and MasterKey. They both compromised 32% and 6% of the victims for touch and gait, respectively. Vanilla-s, K-means++, and MasterKey were the second-best attack on keystroke and mouse biometrics compromising 4% and 28% of the victims, respectively. Against the voice biometric, Hypervolume is the only successful attack in the first attempt, and it compromises 16% of the population. By the fifth attempt, Hypervolume compromises 91% and 86% of the victims in the touch and mouse biometrics. The second-best attack against touch, K-means++, compromised 74% of the victims. Against the mouse biometric, Vanilla-s was able to compromise 61% of the victims. Against gait and voice biometrics, Hypervolume was able to compromise 65% and 34% of the victims, and the second-best attack, K-means++, was able to compromise 28% and 11% of the victims, respectively.

Our evaluations show that Hypervolume performed consistently better than the other attacks against all considered biometrics, except for the first 30 attempts for the voice biometric for KNN.



(a) Touch KNN

(b) Keystroke KNN

(c) Mouse KNN

(d) Gait KNN

(e) Voice KNN

Fig. 5. Performance of attacks against five biometrics trained using KNN classifiers.

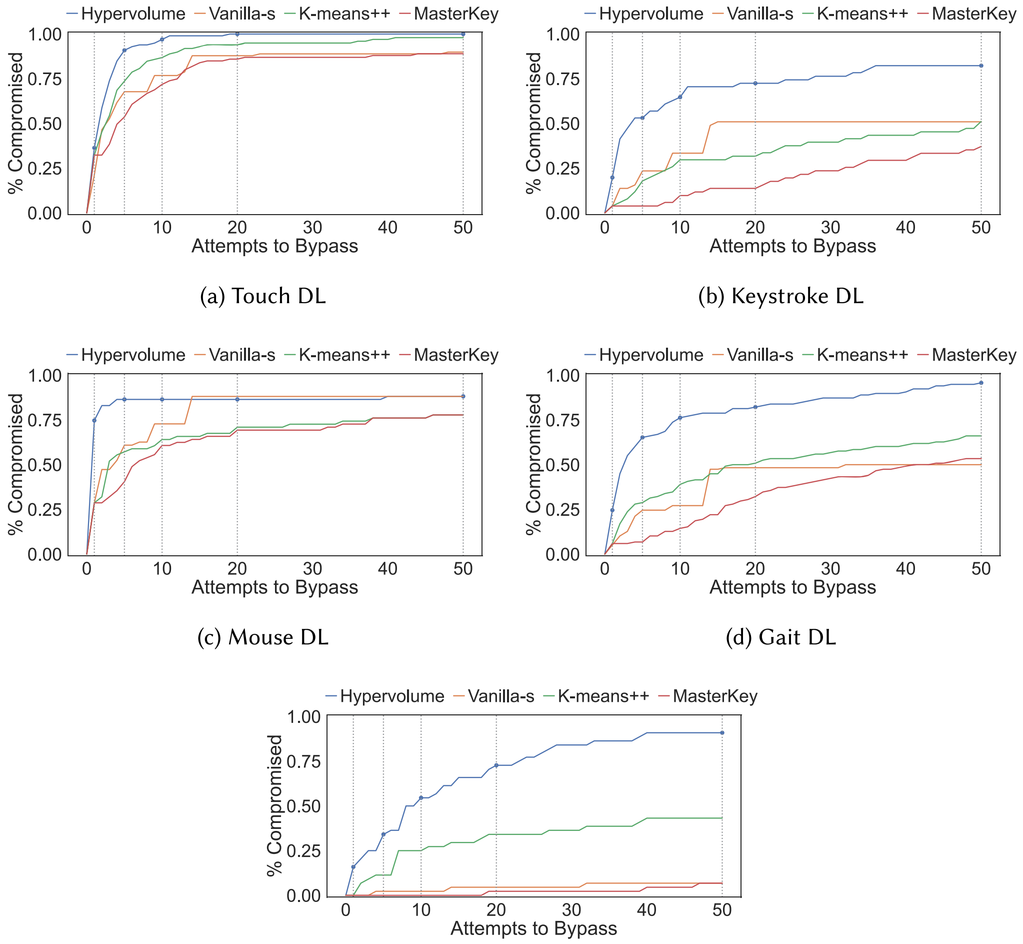
K-means++ was often the second-best attack approach. In the following section, we further compare and contrast the performance of Hypervolume and K-means++.

6.2 Comparison with K-means++

We compare the overall performance of attacks using the **Area Under the Attack Curve (AuAC)** metric, defined in Equation (5). For our calculations, we approximate the integral using the trapezoidal rule [Herman et al. 2016]. AuAC is able to quantify the effectiveness of an attack scenario—a more successful attack will compromise more population in fewer attempts, thus resulting in higher value of AuAC. AuAC is defined by the following equation:

$$AuAC := \frac{1}{n} \int_0^n \text{Percent Population Compromised } d(\text{Attempts to Bypass}), \tag{5}$$

where n is the number of attack attempts.



(a) Touch DL

(b) Keystroke DL

(c) Mouse DL

(d) Gait DL

(e) Voice DL

Fig. 6. Performance of attacks against five biometrics trained using DL classifiers.

Table 3 compares the AuAC for all 20 attack scenarios (five biometrics using four classifiers). Hypervolume has the highest AuAC in all attack scenarios except for the KNN classifier for the voice dataset, where it is the second-best to K-means++. K-means++ is the second-best attack with the second highest AuAC in 11 out of 20 scenarios and the highest AuAC in 1 scenario.

The attack curves can be considered to be cumulative distribution functions (CDFs) representing the distribution of the attacks. It is these underlying distributions that we wish to compare through statistical testing. We perform statistical tests on the attack curves for Hypervolumes and K-means++ (blue and green lines in Figures 3–6). To this end, for the two attacks, we test the following hypotheses:

H_0: The attack curves belong to the same attack distribution.
H_1: The attack curves belong to the different attack distribution.

First, we perform the normality tests on the distribution using D'Agostino and Pearson's test [D'Agostino 1971; D'Agostino and Pearson 1973], which combines skew and kurtosis to

Table 3. AuAC of Attacks

Dataset	Attack	SVM	KNN	RF	DL
Touch	Hypervolume	0.95	0.94	0.93	0.95
	K-means++	**0.93**	0.86	**0.88**	**0.89**
	MasterKey	0.50	0.72	0.62	0.79
	Vanilla-s	0.84	**0.87**	0.84	0.81
Keystroke	Hypervolume	0.66	0.77	0.58	0.71
	K-means++	0.39	0.62	0.45	0.34
	MasterKey	0.17	0.39	0.16	0.20
	Vanilla-s	**0.52**	**0.65**	**0.48**	**0.43**
Mouse	Hypervolume	0.98	0.98	0.97	0.85
	K-means++	0.92	0.86	**0.92**	0.67
	MasterKey	0.55	0.59	0.76	0.64
	Vanilla-s	**0.94**	**0.93**	**0.92**	**0.80**
Gait	Hypervolume	0.80	0.84	0.65	0.80
	K-means++	**0.58**	**0.80**	**0.40**	**0.50**
	MasterKey	0.22	0.19	0.14	0.33
	Vanilla-s	**0.58**	0.61	0.35	0.41
Voice	Hypervolume	0.69	**0.77**	0.68	0.70
	K-means++	**0.51**	0.81	**0.59**	**0.32**
	MasterKey	0.09	0.43	0.00	0.02
	Vanilla-s	0.10	0.60	0.00	0.05

The highest values for each attack and dataset are highlighted. The second highest values for each attack and dataset are shown in **bold**.

produce an omnibus test of normality. For the 40 normality tests, we perform the Bonferroni correction and use the significance level (α) cutoff of $1.25e^{-3}$. The results for normality tests are provided later in Table 6 (see Appendix A), which shows that for most tests (33/40), the D'Agostino and Pearson's test indicates that the distributions are not normal. Therefore, we use a non-parametric equivalent one-way ANOVA [Corder and Foreman 2011; Sidney 1957] and the Kruskal-Wallis test [Kruskal and Wallis 1952]. We perform the Bonferroni correction for the 20 Kruskal-Wallis tests and use the significance level (α) cutoff of $2.5e^{-3}$

Table 4 shows the p values and H statistics for the Kruskal-Wallis tests. It shows that for 17/20 cases, Kruskal-Wallis tests found significant differences in the attack distributions of Hypervolume and K-means++. The three cases for which the null hypothesis was not rejected and both attacks had similar performance include Gait biometric (KNN) and voice biometric (KNN and RF). Figure 5(d) shows that for the gait biometric for KNN, both attack curves for this case are very similar and the AuAC of the curves are also close—0.84 for Hypervolume and 0.8 for K-means++. Figure 5(e) shows that for the voice biometric for KNN, both attack curves for this case are also very similar, but here K-means++ has a slight lead. The AuAC of the curves are also close—0.81 for K-means++ and 0.77 for Hypervolumes. Finally, Figure 4(e) shows that both attack curves are very similar until the 20th attempt for the voice biometric for RF. After the 20th attempt, Hypervolume performed better, and this difference is reflected in the AuAC metric—0.68 for Hypervolume and 0.59 for K-means++.

The reason for the success of Hypervolume and K-means++ is their ability to explore the sample space more effectively compared to Vanilla-s and MasterKey. However, unlike K-means++, Hypervolume does not blindly choose a farther point from the population but chooses a point that is more probable based on the degree of overlap in the clusters, number of unique users in the

Table 4. p Values and H Statistics for the Kruskal-Wallis Test Between Hypervolume and K-means++

Dataset	Classifier	SVM	KNN	RF	DL
Touch	H	10.12	37.33	20.17	38.25
	p	1.46e-03	9.97e-10	7.09e-06	6.23e-10
Keystroke	H	63.01	59.16	55.24	67.09
	p	2.06e-15	1.45e-14	1.07e-13	2.60e-16
Mouse	H	30.31	31.67	22.08	76.50
	p	3.68e-08	1.82e-08	2.61e-06	2.20e-18
Gait	H	49.98	3.16	45.72	60.30
	p	1.55e-12	7.54e-02	1.36e-11	8.14e-15
Voice	H	19.29	0.92	8.71	48.42
	p	1.12e-05	3.38e-01	3.17e-03	3.43e-12

p-Values for which the null hypothesis cannot be rejected are highlighted.

Table 5. Overlap Mean and SD for Different Biometrics

	Touch	Keystroke	Mouse	Gait	Voice
Mean	0.79	0.44	0.75	0.60	0.63
SD	0.02	0.12	0.09	0.06	0.11

cluster, and number of samples in the cluster. This enables Hypervolume to outperform K-means++ in early attempts for most scenarios (refer to Figure 2(d) for a 2D example).

6.3 Discussion

Our experiments are the first to explore and demonstrate the susceptibility of the voice biometric to statistical attacks. Our experiments also show that although several attacks were not proposed against some biometrics (e.g., MasterKey against the touch or mouse biometrics), these attacks achieve good success. Similarly, we observe that some attacks performed better against different classifiers for some biometrics (e.g., MasterKey for the touch biometric for SVM vs. KNN).

Figures 3 through 6 show that different biometrics provide different levels of resilience against statistical attacks. The touch and mouse biometrics perform quite poorly against the attacks and the top-3 attacks compromise 83% of the population within the first 10 attempts, on average. However, for the keystroke, gait, and voice biometrics, only 60% or less of the population is compromised. Table 5 shows the mean population overlaps across different clusters, which explains the attack resilience of these biometrics. It shows that both the touch and mouse biometrics have high mean overlaps compared to the keystroke, gait, and voice biometrics.

We note that the attacks designed to use only statistical properties of the population mean like Vanilla-s and MasterKey are only effective against biometrics with a higher degree of unimodal overlap among users' behavior. Due to the high overlap around the population mean, attack points generated closer to the mean of the population have a higher probability of being successful. Both Vanilla-s and MasterKey only compromise a smaller population against the voice and gait biometrics, where there is less overlap compared to the touch or mouse biometrics. K-means++ performs better than Vanilla-s and MasterKey, as it tries regions farther from the population mean as well. Hypervolume performs consistently better since it not only captures but also ranks the overlapping regions away from the population mean.

We note that using the traditional metrics, some biometrics provide a very similar baseline performance (see Table 2). For example, the EER for the keystroke (mean = 0.11; SD = 0.10) and gait biometric (mean = 0.15; SD = 0.09) for KNN are not too far apart. Similarly, the AUC for keystroke (mean = 0.99; SD = 0.04) and voice (mean = 0.99; SD = 0.01) biometrics are similar for RF. Despite these similarities, the resilience of these biometrics against the attacks is different. Although robust metrics have been proposed that are able to capture the performance of biometric systems better [Eberz et al. 2017; Sugrim et al. 2019], these metrics are not designed to report resilience to statistical attacks. Given the threat posed by statistical attacks, it is prudent to consider a metric similar to "the percentage of the population compromised" and " AuAC" to understand the limitations of these biometrics. This approach will be able to identify biometrics that provide little resistance against statistical attacks for a larger proportion of the population (e.g., the touch and mouse biometrics). This approach also enables the identification of users whose behavioral overlap is high and are more susceptible to statistical attacks. Like the blacklists for knowledge-based authentication systems (e.g., for PINs [Markert et al. 2020] and passwords [Weir et al. 2010]), such "hotspots" of behavioral overlaps could be used to flag the unacceptability of biometrics for certain users. Such a metric will also enable system designers to better understand and defend against threats to their systems from active adversaries. More research needs to be conducted to design a standard metric that captures and communicates the susceptibility of biometrics to statistical attacks.

7 DETECTION MECHANISM FOR STATISTICAL ATTACKS

In this section, we present and evaluate a detection mechanism against statistical attacks.

7.1 Proposed Detection Method

Scope. Our evaluations show that although the evaluated biometrics can defend against adversaries whose behavior does not overlap that of the victim, statistical attacks or attacks where the adversary is actively modifying their behavior pose a serious threat. Adversaries who actively modify their behavior may use an attack discussed in this work to guess the population statistics or randomly modify their behavior (without any prior knowledge of victims' behavior), which may result in similar patterns as the statistical attack (discussed in the following in more detail). Our results showed that victims whose behavior is close to the population mean can be easily bypassed in the first few attempts by all attacks. Furthermore, due to the variations in the intra-user behavior for several biometrics [Khan et al. 2020], it is difficult to create a model of the user's behavior that rejects overlapping samples from other users. Therefore, it is quite challenging to detect attacks where the victim's behavior is closer to the population mean and is compromised in the first attempt. When the behavior of the victim is farther from the population mean, attacks like MasterKey and Vanilla-s can be effortlessly detected using rule-based techniques. In the case of MasterKey, subsequent rejected feature vectors will have very similar distance between them, and they will most likely be not near the centroid(s) of a user's data. Vanilla-s will be producing feature vectors in and around a particular area. Both behaviors are different from a legitimate user's feature vectors, which would not always be concentrated in a particular region farther from the user's behavior.

Our proposed detection method needs two or three samples for detecting attacks. Since Hypervolume attack was able to compromise 100%, 98.34%, 94.94%, and 90% of the population for SVM, KNN, RF, and DL classifiers, respectively, in just two attempts, we choose not to evaluate our detection method on this biometric due to the lack of data. In this work, we focus on building a detection method against more potent attacks including Hypervolume and K-means++.

Intuition. Our assumption is that the attackers prefer an attack that requires a lower number of attempts to bypass [Acar et al. 2020; Khan et al. 2018; Negi et al. 2018], since a higher failure rate could be easily detected using a predefined threshold (similar to the throttling of failed password attempts). We observe that attacks with this desirable property explore the sample space more effectively by exploring a region of data farther from the last submitted attack sample in case of a failure. Similarly, an attacker who is trying to actively defeat the biometric system, by randomly modifying their behavior after every failed attempt (rejected samples), moves farther away from the region that captures their current attempt. This differs from the normal behavior of the user or an attacker who is not actively trying to evade the biometric defence, where the behavior across subsequent attempts does not vary greatly and is closer to the region(s) or cluster(s) containing the behavior of the user or the attacker, respectively. For rejected samples from the user (false rejects), the probability that these samples are farther from the region(s) or cluster(s) containing the behavior of the user and subsequent rejected samples are farther from each other should be low. Our detection method against statistical attacks exploits this observation.

Our proposed detection method continuously operates in the background and takes as input two classifier scores and corresponding samples (i.e., feature vectors). Note that the proposed detection method may be configured to operate only on the samples that were rejected, but since the numbers of *consecutively* rejected samples for users are low for voice, keystroke, and gait in our datasets, we test our defense as continuously operating in the background. For the two samples (S_1, S_2) for a user with centroid of the user, C, the detection method computes the attack probability, P_A, using the following equation:

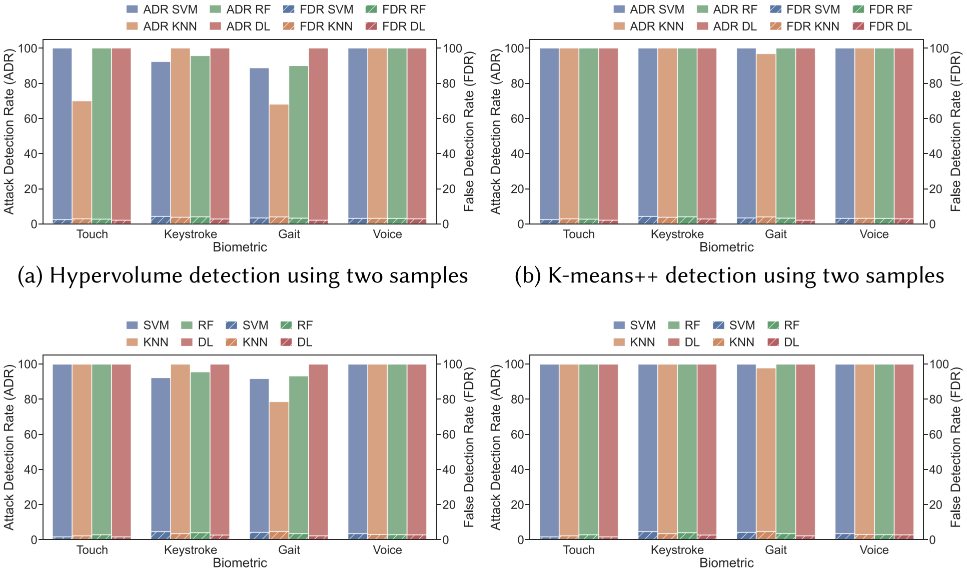
$$\begin{aligned} P_A = Dist(S_1, C)w_1 + Dist(S_2, C)w_2 + Dist(S_1, S_2)w_3 \\ + (win_{size} - (Clsfr(S_1) + Clsfr(S_2)))w_4, \end{aligned} \tag{6}$$

where $Dist$ is the cosine distance, $Clsfr(S_x)$ is the binary outcome of the classifier ('1' indicates that the sample belongs to the user's class), w_x are the weights, and win_{size} is the number of samples in the sliding window. Each distance and classifier score is scaled using weights so that their value is between 0 and 1. This attack probability is computed over a sliding window of two or three samples. For a window of three samples, like Equation (6), we compute a weighted sum of all pairwise distances between the three samples, the distance between the three sample and the centroid, and classifier scores for the three samples.

Evaluation and Results. For statistical attack samples, we focus on users in each dataset who require two or more attempts to compromise by the evaluated attacks. We split each users' data into two disjoint groups: 80% of the data is used to estimate the thresholds for the attack probability, whereas the remaining data is used to compute the FDR. For our experiments, we use the 97th percentile of the calculated score as the threshold. If the observed score is above this threshold, samples are tagged as possible statistical attacks.

Figure 7 shows the **Attack Detection Rate (ADR)** and FDR against the two attacks considered for the evaluation of the detection method proposed in this work. ADR captures the correctly detected proportion of statistical attacks, and FDR captures the proportion of the user's samples misclassified as statistical attack samples.

For Hypervolume, Figure 7(a) shows that for two samples, the proposed detection method can provide a high ADR (85% or higher) with a low FDR (5% or lower) against all classifiers but KNN for the gait biometric. The ADR for KNN is quite low for the touch and gait biometrics—70% and 68%, respectively. With three samples, we see an increase in the ADR for touch from 70% to 100%; for gait, it increases from 68% to 78% with similar FDR (2.4% and 4.8%, respectively) (see Figure 7(c)). The performance of the detection method on the keystroke dataset on the SVM classifier remained the same for both using two or three points, and 92% of the attacks were detected. Compared with the other efficient attack, K-means++ (see the following), we note that detection of Hypervolume is more challenging. This is because Hypervolume does not submit the farthest region from the



(a) Hypervolume detection using two samples (b) K-means++ detection using two samples

(c) Hypervolume detection using three samples (d) K-means++ detection using three samples

Fig. 7. Attack detection rate of the proposed defense using a window of two and three samples.

current attack sample and instead prioritizes which region to submit next. Therefore, although likely, the next attack sample may not surpass the chosen threshold.

Figure 7(b) shows that the proposed detection method provides a near perfect ADR (97% or higher) with a low FDR (5% or lower) against K-means++ with only two samples. K-Means++ is easier to detect since it mostly tries the next sample that is farther away from the previous failed sample. With three samples, K-means++ attack was detected with an ADR of 98% or higher similar FDR (Figure 7(d)). Against the two detrimental attacks that explore the search space more efficiently and pose a serious threat to biometrics, on average, our detection method can achieve 98% ADR with only 3.2% FDR using three samples.

We note that an adversary can evade our detection method if a rule-based defense is not complimented by submitting new attack samples that have smaller distance from the previous samples (i.e., like MasterKey, explores the attack space with a smaller step size). Although this approach is difficult to flag using our defense, it requires a larger number of attempts to evade the biometric system, thereby increasing the chances of getting detected using simpler techniques such as rate limiting. Finally, although our detection method does not protect victims whose behavior overlaps that of the population mean, our approach provides a method for security practitioners to preemptively identify vulnerable users for a biometric and flag their unsuitability for the biometric.

8 CONCLUSION

We propose an attack that uses hypervolumes to capture overlap patterns for different biometrics across population. We show that our attack can evade a diverse set of biometrics and performs better when compared with other state-of-the-art attack methods. More specifically, our attack can compromise a much higher proportion of the population with fewer attempts than the other attacks. Furthermore, we propose a simple detection mechanism that can detect attacks that pose a serious threat to these systems (i.e., attacks that cannot be throttled using simple rate limiting techniques). For victims whose behavior is farther from the population mean, our detection mechanism can use two to three samples to detect a statistical attack with high accuracy. Our work also exposes a previously undiscovered weakness of the voice biometric and highlights the need to evaluate biometric authentication systems using metrics that capture the performance against statistical attacks.

APPENDIX

A NORMALITY TEST

Table 6. Normality Test Results for Attack Curves

Dataset	Attack	Classifier	SVM	KNN	RF	DL
Touch	Hypervolume	H	70.48	68.12	80.28	73.65
		p	4.97e-16	1.61e-15	3.70e-18	1.02e-16
	K-means++	H	67.26	28.91	41.45	50.40
		p	2.48e-15	5.28e-07	9.98e-10	1.14e-11
Keystroke	Hypervolume	H	50.69	76.92	40.84	35.18
		p	9.84e-12	1.99e-17	1.35e-09	2.29e-08
	K-means++	H	8.89	21.95	40.35	11.00
		p	1.17e-02	1.71e-05	1.73e-09	4.09e-03
Mouse	Hypervolume	H	112.25	112.25	103.02	80.82
		p	4.23e-25	4.23e-25	4.27e-23	2.82e-18
	K-means++	H	67.07	24.67	59.20	39.14
		p	2.73e-15	4.39e-06	1.40e-13	3.17e-09
Gait	Hypervolume	H	33.36	39.86	22.69	36.17
		p	5.69e-08	2.21e-09	1.18e-05	1.40e-08
	K-means++	H	50.18	33.04	33.87	13.90
		p	1.27e-11	6.69e-08	4.41e-08	9.61e-04
Voice	Hypervolume	H	10.17	13.10	9.50	9.73
		p	6.19e-03	1.43e-03	8.66e-03	7.73e-03
	K-means++	H	6.33	16.72	14.76	15.15
		p	4.23e-02	2.34e-04	6.24e-04	5.12e-04

H is the statistic $H = s^2 + k^2$, where s is the z-score returned by skew test and k is the z-score returned by kurtosis test. p is the two-sided χ^2 probability. p-values for which null-hypothesis cannot be rejected are highlighted.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*. 265–283.

Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Kemal Akkaya. 2020. A usable and robust continuous authentication framework using wearables. *IEEE Transactions on Mobile Computing* 20, 6 (2020), 2140–2153.

Ahmed Awad E. Ahmed and Issa Traore. 2007. A new biometric technology based on mouse dynamics. *IEEE Transactions on Dependable and Secure Computing* 4, 3 (2007), 165–179.

Lucas Ballard, Fabian Monrose, and Daniel P. Lopresti. 2006. Biometric authentication revisited: Understanding the impact of wolves in sheep's clothing. In *Proceedings of the USENIX Security Symposium.*

BehavioSec. 2021. Continuous Authentication Solutions. Retrieved September 1, 2021 from https://www.behaviosec.com/.

Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. 2002. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security* 5, 4 (2002), 367–397.

Benjamin Blonder. 2018. Hypervolume concepts in niche-and trait-based ecology. *Ecography* 41, 9 (2018), 1441–1455.

Benjamin Blonder and David J. Harris. 2019. Hypervolume: High Dimensional Geometry and Set Operations Using Kernel Density Estimation, Support Vector Machines, and Convex Hulls (Version 2.0.12). Retrieved November 25, 2022 from https://CRAN.R-project.org/package=hypervolume.

Benjamin Blonder, Cecina Babich Morrow, Brian Maitner, David J. Harris, Christine Lamanna, Cyrille Violle, Brian J. Enquist, and Andrew J. Kerkhoff. 2018. New approaches for delineating *n*-dimensional hypervolumes. *Methods in Ecology and Evolution* 9, 2 (2018), 305–319.

Nikolaos V. Boulgouris, Dimitrios Hatzinakos, and Konstantinos N. Plataniotis. 2005. Gait recognition: A challenging signal processing technology for biometric identification. *IEEE Signal Processing Magazine* 22, 6 (2005), 78–90.

Marcelo Luiz Brocardo, Issa Traore, and Isaac Woungang. 2014. Toward a framework for continuous authentication using stylometry. In *Proceedings of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications*. IEEE, Los Alamitos, CA, 106–115.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, et al. 2013. API design for machine learning software: Experiences from the scikit-learn project. In *Proceedings of the European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*.

Chris Burt. 2018. Biometrics-Secured Voice Banking with Amazon Alexa Now Available from Two Canadian Credit Unions. Retrieved July 1, 2021 from https://www.biometricupdate.com/201811/biometrics-secured-voice-banking-with-amazon-alexa-now-available-from-two-canadian-credit-unions.

Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. VoxCeleb2: Deep speaker recognition. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'18)*.

Nathan L. Clarke and S. M. Furnell. 2007. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security* 6, 1 (2007), 1–14.

Gregory W. Corder and Dale I. Foreman. 2011. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach. Wiley.*

Ralph D'Agostino. 1971. An omnibus test of normality for moderate and large sample sizes. *Biometrika* 58, 34 (1971), 1–348.

Ralph D'Agostino and Egon S. Pearson. 1973. Tests for departure from normality. Empirical results for the distributions of b^2 and $\sqrt{b}$. *Biometrika* 60, 3 (1973), 613–622.

Simon Eberz, Giulio Lovisotto, Andrea Patane, Marta Kwiatkowska, Vincent Lenders, and Ivan Martinovic. 2018. When your fitness tracker betrays you: Quantifying the predictability of biometric features across contexts. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP'18)*. IEEE, Los Alamitos, CA, 889–905.

Simon Eberz, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. 2017. Evaluating behavioral biometrics for continuous authentication: Challenges and metrics. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, New York, NY, 386–399.

Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security* 8, 1 (2013), 136–148.

Lex Fridman, Steven Weber, Rachel Greenstadt, and Moshe Kam. 2016. Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. *IEEE Systems Journal* 11, 2 (2016), 513–521.

Davrondzhon Gafurov, Einar Snekkenes, and Patrick Bours. 2007. Spoof attacks on gait authentication system. *IEEE Transactions on Information Forensics and Security* 2, 3 (2007), 491–502.

Yang Gao, Rita Singh, and Bhiksha Raj. 2018. Voice impersonation using generative adversarial networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. IEEE, Los Alamitos, CA, 2506–2510.

Rosa González Hautamäki, Tomi Kinnunen, Ville Hautamäki, and Anne-Maria Laukkanen. 2015. Automatic versus human speaker verification: The case of voice mimicry. *Speech Communication* 72 (2015), 13–31.

Edwin Herman, Gilbert Strang, William Radulovich, Erica A. Rutter, David Smith, Kirsten R. Messer, Alfred K. Mulzet, Nicoleta Virginia Bila, et al. 2016. *Calculus: Volume 2*. XanEdu Publishing.

G. Evelyn Hutchinson. 1957. *A Treatise on Liminology*. Wiley.

Anil K. Jain, Arun Ross, and Salil Prabhakar. 2004. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 1 (2004), 4–20.

Zach Jorgensen and Ting Yu. 2011. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer, and Communications Security*. ACM, New York, NY.

Robert R. Junker, Jonas Kuppler, Arne C. Bathke, Manuela L. Schreyer, and Wolfgang Trutschnig. 2016. Dynamic range boxes—A robust nonparametric approach to quantify size and overlap of *n*-dimensional hypervolumes. *Methods in Ecology and Evolution* 7, 12 (2016), 1503–1513.

Hassan Khan, Urs Hengartner, and Daniel Vogel. 2016. Targeted mimicry attacks on touch input based implicit authentication schemes. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, New York, NY.

Hassan Khan, Urs Hengartner, and Daniel Vogel. 2018. Augmented reality-based mimicry attacks on behaviour-based smartphone authentication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, New York, NY.

Hassan Khan, Urs Hengartner, and Daniel Vogel. 2020. Mimicry attacks on smartphone keystroke authentication. *ACM Transactions on Privacy and Security* 23, 1 (2020), 1–34.

Kevin S. Killourhy and Roy A. Maxion. 2009. Comparing anomaly-detection algorithms for keystroke dynamics. In *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks.* IEEE, Los Alamitos, CA.

Achim Klenke. 2007. *Probability Theory: A Comprehensive Course.* Springer Science & Business Media.

William H. Kruskal and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* 47, 260 (1952), 583–621.

Justin Lee. 2016. NuData More Than Doubles Behavioral Transaction Volume. Retrieved September 1, 2021 from http://www.biometricupdate.com/201605/nudata-security-more-than-doubles-behavioral-transaction-volume.

Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable reauthentication for smart phones. In *Proceedings of the 20th Network and Distributed System Security Symposium.*

Dachuan Liu, Bo Dong, Xing Gao, and Haining Wang. 2015. Exploiting eye tracking for smartphone authentication. In *Proceedings of the International Conference on Applied Cryptography and Network Security.* 457–477.

Zongyi Liu and Sudeep Sarkar. 2006. Improved gait recognition by gait dynamics normalization. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6 (2006), 863–876.

Giulio Lovisotto, Simon Eberz, and Ivan Martinovic. 2020. Biometric backdoors: A poisoning attack against unsupervised template updating. In *Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P'20).* IEEE, Los Alamitos, CA, 184–197.

Anthony Maeder, Clinton Fookes, and Sridha Sridharan. 2004. Gaze based user authentication for personal computer applications. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video, and Speech Processing.* IEEE, Los Alamitos, CA, 727–730.

Philipp Markert, Daniel V. Bailey, Maximilian Golla, Markus Dürmuth, and Adam J. Aviv. 2020. This pin can be easily guessed: Analyzing the security of smartphone unlock pins. In *Proceedings of the IEEE Symposium on Security and Privacy.* IEEE, Los Alamitos, CA.

Stephen Mayhew. 2016. Nationwide Mobile Banking App Uses Behavioral Biometrics. Retrieved September 1, 2016 from http://www.biometricupdate.com/201604/nationwide-mobile-banking-app-uses-behavioral-biometrics.

Manar Mohamed, Babins Shrestha, and Nitesh Saxena. 2016. SMASheD: Sniffing and manipulating android sensor data for offensive purposes. *IEEE Transactions on Information Forensics and Security* 12, 4 (2016), 901–913.

Fabian Monrose and Aviel Rubin. 1997. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM Conference on Computer and Communications Security.* 48–56.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. 2020. VoxCeleb: Large-scale speaker verification in the wild. *Computer Speech & Language* 60 (2020), 101027.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. VoxCeleb: A large-scale speaker identification dataset. In *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH'17).*

Parimarjan Negi, Prafull Sharma, Vivek Jain, and Bahman Bahmani. 2018. K-means++ vs. behavioral biometrics: One loop to rule them all. In *Proceedings of the 25th Network and Distributed System Security Symposium.*

Saurabh Panjwani and Achintya Prakash. 2014. Crowdsourcing attacks on biometric systems. In *Proceedings of the 10th Symposium on Usable Privacy and Security.*

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20 (1987), 53–65.

Samsung SDS. 2021. Nexsign: Behavioral Biometrics for Continuous Frictionless Identity Authentication. Retrieved September 1, 2021 from https://www.samsungsds.com/us/behavioral/biometrics.html.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a " kneedle"in a haystack: Detecting knee points in system behavior. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops.* IEEE, Los Alamitos, CA, 166–171.

Manuela Schreyer, Robert R. Junker, Wolfgang Trutschnig, Jonas Kuppler, Arne Bathke, Judith H. Parkinson, and Raoul Kutil. 2018. dynRB: Dynamic Range Boxes (Version 0.15). Retrieved November 25, 2022 from https://CRAN.R-project.org/package=dynRB.

Abdul Serwadda and Vir V. Phoha. 2013a. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM Transactions on Information and System Security* 16, 2 (2013), 8.

Abdul Serwadda and Vir V. Phoha. 2013b. When kids' toys breach mobile phone security. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY.

Abdul Serwadda, Vir V. Phoha, and Ankunda Kiremire. 2011. Using global knowledge of users' typing traits to attack keystroke biometrics templates. In *Proceedings of the 13th ACM Multimedia Workshop on Multimedia and Security*.

Abdul Serwadda, Vir V. Phoha, Zibo Wang, Rajesh Kumar, and Diksha Shukla. 2016. Toward robotic robbery on the touch screen. *ACM Transactions on Information and System Security* 18, 4 (2016), 1–25.

Siegel Sidney. 1957. Nonparametric statistics for the behavioral sciences. *Journal of Nervous and Mental Disease* 125, 3 (1957), 497.

Kesar Singh and Minge Xie. 2008. Bootstrap: A statistical method. Unpublished manuscript. Rutgers University.

Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S. Balagani. 2015. HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security* 11, 5 (2015), 877–892.

Valeriu-Daniel Stanciu, Riccardo Spolaor, Mauro Conti, and Cristiano Giuffrida. 2016. On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. In *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy*. 105–112.

Øyvind Stang. 2007. *Gait Analysis: Is it Easy to Learn to Walk Like Someone Else?* Master's thesis. Gjøvik University College, Norway.

Shridatt Sugrim, Can Liu, Meghan McLean, and Janne Lindqvist. 2019. Robust performance metrics for authentication systems. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS'19)*.

Heidi K. Swanson, Martin Lysy, Michael Power, Ashley D. Stasko, Jim D. Johnson, and James D. Reist. 2015. A new probabilistic method for quantifying *n*-dimensional ecological niches and niche overlap. *Ecology* 96, 2 (2015), 318–324.

Chee Meng Tey, Payas Gupta, and Debin Gao. 2013. I can be you: Questioning the use of keystroke dynamics as biometrics. In *Proceedings of the Annual Network and Distributed System Security Symposium*.

Hoang Minh Thang, Vo Quang Viet, Nguyen Dinh Thuc, and Deokjai Choi. 2012. Gait identification using accelerometer on mobile phone. In *Proceedings of the International Conference on Control, Automation, and Information Sciences*. IEEE, Los Alamitos, CA.

Aad W. Van der Vaart. 2007. *Asymptotic Statistics*. Vol. 3. Cambridge University Press.

Sergei Vassilvitskii and David Arthur. 2006. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1027–1035.

Sébastien Villéger, Norman W. H. Mason, and David Mouillot. 2008. New multidimensional functional diversity indices for a multifaceted framework in functional ecology. *Ecology* 89, 8 (2008), 2290–2301.

Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*. 162–175.

Hui Xu, Yangfan Zhou, and Michael R. Lyu. 2014. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Proceedings of the Symposium on Usable Privacy and Security*.

Benjamin Zi Hao Zhao, Hassan Jameel Asghar, and Mohamed Ali Kaafar. 2020. On the resilience of biometric authentication systems against random inputs. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS'20)*.

Nan Zheng, Aaron Paloski, and Haining Wang. 2011. An efficient user verification system via mouse movements. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*.

Tiantian Zhu, Lei Fu, Qiang Liu, Zi Lin, Yan Chen, and Tieming Chen. 2020. One cycle attack: Fool sensor-based personal gait authentication with clustering. *IEEE Transactions on Information Forensics and Security* 16 (2020), 553–568.

Qin Zou, Yanling Wang, Qian Wang, Yi Zhao, and Qingquan Li. 2020. Deep learning-based gait recognition using smartphones in the wild. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3197–3212.

Received 10 March 2022; revised 21 August 2022; accepted 31 October 2022