Unsure How to Authenticate on Your VR Headset? Come on, Use Your Head!

Tahrima Mustafa, Richard Matovu, Abdul Serwadda, Nicholas Muirhead
Texas Tech University
Lubbock, Texas
{tahrima.mustafa,richard.matovu,abdul.serwadda,nicholas.muirhead}@ttu.edu

ABSTRACT

For security-sensitive Virtual Reality (VR) applications that require the end-user to enter authenticatioan credentials within the virtual space, a VR user's inability to see (potentially malicious entities in) the physical world can be discomforting, and in the worst case could potentially expose the VR user to visual attacks.

In this paper, we show that the head, hand and (or) body movement patterns exhibited by a user freely interacting with a VR application contain user-specific information that can be leveraged for user authentication. For security-sensitive VR applications, we argue that such functionality can be used as an added layer of security that minimizes the need for entering the PIN. Based on a dataset of 23 users who interacted with our VR application for two sessions over a period of one month, we obtained mean equal error rates as low as 7% when we authenticated users based on their head and body movement patterns.

CCS CONCEPTS

• **Security and privacy → Authentication**; *Biometrics*;

KEYWORDS

Virtual Reality, Continuous Authentication, Behavioral Biometrics, Head Movement Patterns

ACM Reference Format:
Tahrima Mustafa, Richard Matovu, Abdul Serwadda, Nicholas Muirhead. 2018. Unsure How to Authenticate on Your VR Headset? Come on, Use Your Head!. In *IWSPA'18: 4th ACM International Workshop on Security And Privacy Analytics, March 19–21, 2018, Tempe, AZ, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3180445.3180450

1 INTRODUCTION

Following decades of promise, Virtual Reality (VR) is finally on the path to becoming one of the next big things in technology. For problems spanning domains such as education [2], medical care [11], advertisement [6], manufacturing [1] and shopping [21] to mention but a few, VR is at the heart of new innovative applications that have already begun to have a game changing impact.

As VR practitioners continue to innovate however, one weakness is increasingly becoming apparent — very little research, has explored the security and privacy issues facing the VR revolution. Because user interactions in a VR setting are drastically different from how users interact with devices in the physical world, certain well understood threats take on a completely different dimension in the VR space, making traditional defense mechanisms imperfect and in some cases impossible to apply to VR.

Take the case of user authentication for instance. For strong security, mobile devices (e.g., smart phones) require the user to re-enter the PIN periodically (e.g., when the screen goes dormant for a minute or so). This same functionality is implemented at app level for security-sensitive apps — e.g., shopping apps often require the user to re-enter the PIN when the user makes the final "buy" operation following a period of browsing. For a user working in the physical space (e.g., on a smart-phone), such PIN re-entry is straightforward and incurs minimal inconvenience.

In a VR setting however, it is not obvious how similar functionality can be achieved while guaranteeing both the usability of the application and the secrecy of the PIN. The most natural way to periodically re-enter the PIN on a VR app is to have the user type the PIN within the virtual space. For example, using a pointer device held in the hands, the user can point at different locations representing the PIN in the virtual space. The challenge with this option is that it has the potential to make the PIN extremely vulnerable to visual attacks.

For example, an adversary who has knowledge of the PIN entry screen structure within the VR device could very easily decode hand movements from a video recording to infer PIN inputs (see example attack that decodes hand movements to decode the PIN [24]). If the authentication process is centered around a pattern lock (say, on the main authentication screen), inference of the pattern even becomes much easier since the attacker only needs to track the hand movement trajectory and requires no exquisite knowledge of the input screen design (see example attack that decodes hand movements to decode the pattern lock [27]).

Because a VR user's view is completely blocked from the physical world, there is very little leverage to obfuscate the PIN/pattern entry process from an adversary seeking to launch these kinds of attacks. If the application in question is highly sensitive, security-cautious users will likely be skeptical about "blindly" entering and re-entering their PINs/patterns in such a setting, especially when other persons are in close proximity (e.g., located in the same room).

In the interest of security, a more conservative option would perhaps be to remove the VR device from the head and enter the PIN on some physical interface on the device itself or on a device paired with the VR device. This way, the user is fully aware of

IWSPA'18, March 19–21, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5634-3/18/03…$15.00
https://doi.org/10.1145/3180445.3180450

the environment and is hence better placed to mask the PIN entry process from potential adversaries. However, having to perpetually move the device on and off the head for PIN re-entry is not only inconvenient, but also potentially strenuous to the eyes.

In this paper, we argue that the behavioral patterns exhibited by a user traversing the virtual space might contain identifying cues that provide some measure of confidence about the person using the VR device. If these cues can be reliably captured, they could be used to provide an extra layer of security for sensitive processes carried out in the VR space. For example, given a user entering (and re-entering) the PIN within the virtual space, the system could be designed to authenticate not only the PIN itself, but also the behavioral patterns associated with PIN entry. Beyond PIN entry, such patterns could provide some mechanism of periodic authentication during the user's traversal of the virtual space. One example of the behavioral patterns which could be leveraged this way are the various body and head movements exhibited while the user moves from one point to the next in the virtual space. Using inertial and orientation sensors embedded in the VR device, these movements can be recorded and used to build a biometric profile for the user in question.

This paper studies the hypothesis presented in the previous paragraph. In particular, we investigated whether the head and body movement patterns exhibited by a user traversing the VR space could be used to divulge information about the identity of the user. We designed a VR app in which users moved from one position to the next in the virtual space by moving their body and (or) head. Using accelerometer and gyroscope sensor data captured during these movements, we built user profiles and carried out authentication on the users.

The main **contributions of our work** are described below:

(1) *A new biometric authentication modality for VR*: We present head and body movement patterns as a biometric authentication modality in VR. To illustrate this modality, we build a simple VR game in which users follow the path of a ball in the virtual space, and leverage this game to design the authentication framework and study its performance.
(2) *Performance evaluation of the biometric system*: Using data collected from 23 users who interacted with our application in two sessions over a period of one month, we evaluate the authentication performance of the VR biometric framework. Across the full user population, we show our method to attain a mean equal error rate of about 7% in our best configuration. In a user-level performance analysis, we find that a small proportion of the user population disproportionately contributes to the mean equal error rate, which in turn implies that a selective enrollment policy could significantly lower the equal error rate.
(3) *VR authentication feature-set*: We define and extract a catalog of 178 features from the accelerometer and gyroscope sensors and study their performance under our VR authentication framework. The performance of the features used in our work should guide feature formulations for future research exploring behavioral biometric authentication in VR.

Road-map: The rest of the paper is organized as follows. In Section 2, we present our threat model and discuss examples of security-sensitive VR applications that might benefit from our method. We discuss related work in Section 3 and present our data collection experiments in Section 4. We finally present our results in Section 5 and our conclusions in Section 6.

2 THREAT MODEL AND EXAMPLE VR APPS

Besides potentially improving the usability of security-sensitive VR applications (recall scenarios given in Section 1, Page 1), our method offers a layer of defense against the so called *lunch time attack* [5]. In the context of VR, the attack involves a malicious entity gaining access to an unlocked VR device that belongs to another user (e.g., when the legitimate user has stepped away from her desk for lunch [5]). The attacker's motivations may range from just viewing private information (e.g., an employer might be interested in learning what items an employee just bought from an online VR store) to stealing highly sensitive data saved in the app (e.g., credit card numbers).

After a series of uncharacteristic head and body movements during the traversal of the virtual space, our method will lock the device or prompt the attacker to prove their identity via an alternative modality. Figure 1 gives a high level view of how such an authentication system might be designed. While the user traverses the virtual space, sensor data produced is segmented into windows, and classification decisions made for each window. If a certain threshold number of windows indicate that the user is an impostor, then the user is prompted to use a secondary modality (e.g., via a static method such as the PIN) before access is resumed. The size of the windows and the number and nature of uncharacteristic movements triggering the secondary authentication mechanism would in practice be tuned to strike a balance between usability and security.

Below, we give two concrete examples of VR apps which might benefit from our methods.

Example App #1: The family of VR (or AR) shopping apps provide a good example of civilian-targeted apps that would benefit from enhanced security mechanisms such as ours. The Chinese e-commerce site, Alibaba, is already test-driving one such app (see *Buy+* app [4, 10]), while Amazon is said to be developing its own app [26]. These apps enable the customer to browse through items in a virtual mall in a way that mimics how they would traverse a physical mall. Using head, hand and (or) body movements, the user is able to move from one point to the next in this virtual mall and to select and buy products using payment details already saved in the app. These motion patterns could be used to provide some measure of confidence about who is traversing the virtual space, which would in turn potentially minimize the frequency of password entry for user authentication.

Example App #2: For reasons ranging from earning game credits to simply getting personal satisfaction, unscrupulous gamers are well-known to engage in many forms of cheating [3]. In Massively Multi-player Online Games (MMOG), one form of cheating is through one player impersonating a different player [3, 9]. Assuming a VR-based game, if a user's head, hand and (or) body movement patterns while playing the game can be exploited to provide some

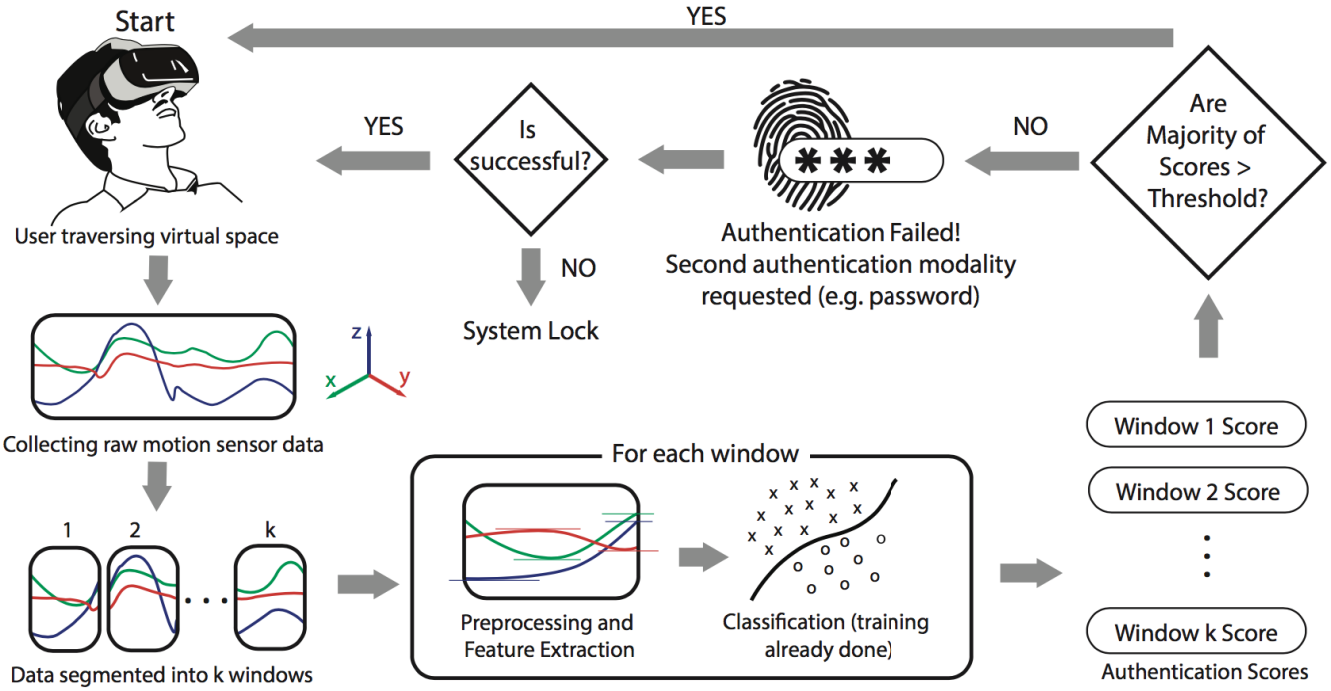

Figure 1: High level overview of VR authentication mechanism using head and body movement patterns.

measure of confidence about who is playing the game, this would minimize the likelihood of impersonation.

3 RELATED WORK

Active Authentication: There is a growing body of research on Active Authentication (AA) methods such as the one proposed in this paper. The principal motivation behind AA is that after a password is entered at login-time, a computing system has no other way to determine whether the individual using the system is the same individual who logged in at the start of a session. AA techniques tackle this challenge by leveraging attributes of the user's interaction with the system to *continuously* authenticate the user. The vast majority of AA systems in the literature use behavioral patterns such as typing [17, 20], gait [14, 25], touch [8, 23], eye movement patterns [5, 18] and brain signal patterns [19, 22] while a smaller segment of research has explored physical biometric modalities such as the face [7, 15, 16].

While physical biometrics such as face are more accurate relative to behavioral biometrics, they are in general more intrusive to the end user. For example, for a smart phone user's face to be continuously authenticated, the user's face has to always be within the field of view of the phone camera, a requirement which might be intrusive for certain tasks and (or) users. On the other hand to authenticate the user based behavioral traits such as typing, gait or touch patterns, there is no such restrictive requirement since the biometric data used comes naturally from the typing, walking or touch process being authenticated by the device.

Our method adds to the behavioral sub-domain within the AA family. By virtue of relying on body movement patterns that naturally emerge from the user's interaction with the VR device, the method is largely non-intrusive, a key property of modalities in this family.

Head Movement Patterns: Two recent AA works (see [13, 28]) that used head movement patterns as a biometric relate closely to our work. In both works, head movement patterns were captured using motion and orientation sensors embedded in Google glasses. In the work by Li *et al.* (see full paper [13] and demo description [12]), users were authenticated based on how they nodded their heads in response to music. Depending on how long the nodding went on, the authors were able to obtain mean EERs of between 4.43% and 24.94% based on a dataset of 30 subjects. In the work by Yi *et al.* [28], users were authenticated based on how they executed a set of six head gestures whose shapes included a circle, triangle, square and three kinds of lines. In order to precisely match the shapes of the gestures, participants used their nose as a pointer (or pen) — e.g., to execute a circular gesture, participants used the nose to trace out the shape of a circle. Based on a dataset of 18 users, Yi *et al.* obtained authentication accuracies as high as 92%. By virtue of analyzing head movement patterns as a biometric, these works have elements in common with our work. That said, the two lines of work have several significant differences from our work that are summarized below.

(1) *Login-time vs continuous authentication*: The static, predefined, gestures used by Yi *et al.* and Li *et al.* target a one-time authentication scenario where a user executes a special gesture before being

allowed access to a resource. For the kind of continuous authentication application targeted by our work, such predefined gestures would not only be intrusive, but would also potentially underestimate the authentication error rates that could be obtained in a real VR app. For example, having a user execute a square-shaped gesture every few minutes for the sake of security would significantly dampen the user's experience in the VR space. Further, after multiple executions of the gesture, the user might develop some consistent technique of implementing it, significantly lowering the error rates. This in turn implies that error rates seen in an experiment based on static gestures would not realistically paint the picture of what might be seen in a continuous authentication setting where the user would not have the benefit of being authenticated based on a memorized action.

To closely model the continuous authentication setting, our work did not restrict users to some fixed, predefined and well-known, library of gestures. Rather, we had users freely make body and head movements in response to random events happening within the virtual space.

(2) *Virtual domain vs physical domain*: The experiments in [12, 13, 28] are conducted in the physical domain, and thus provide little or no insights into the problem of authenticating a user based on interactions undertaken in the virtual space. From the ground up, our work is tailored to the virtual space and is the first to conduct a continuous authentication experiment in VR.

(3) *Body parts forming biometric footprint*: A user's biometric footprint in our experiment was tailored to movements of the VR pointer, which in turn were accomplished through head movements, or other body movements (e.g., movements about the chest area, waist area or even the whole upper body turning about the seat).

Just like our work, this line of work is motivated by the need for authentication in VR settings. However, the work is focused on login-time authentication (via password and PIN) and does not address the continuous authentication problem studied in our research.

4 EXPERIMENT DESIGN AND DATA PROCESSING

4.1 Mechanism of VR App Used for our Experiments

To evaluate the head and body movement biometric, we built a simple VR game in which the movement of a ball was used to prompt users to move from one point to another in the virtual space. The app operates as follows: While a user is at point A in the VR space, a ball becomes visible at another point B. The user then heads towards point B. On arriving at point B, another ball becomes visible at some random point C. The user then moves towards C (see Figure 2 for illustration). The process continues until a user covers 25 balls. Movement towards a ball is attained by moving the VR pointer towards the general direction of the ball. In turn the pointer is moved by nudging the VR headset in a given direction, which can be done by turning the head, bending the body in different directions or even turning on one's seat from one direction to another.

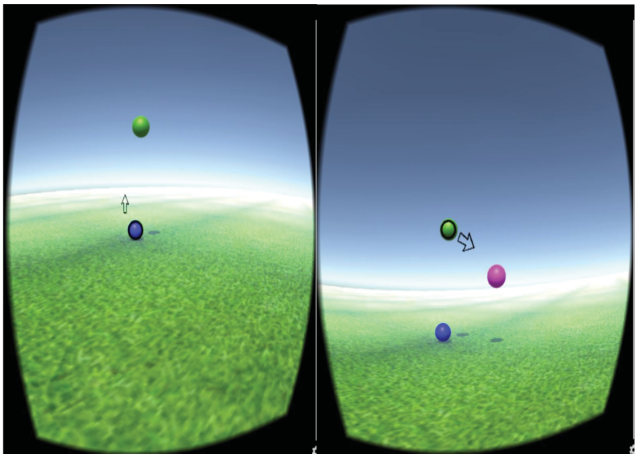

Figure 2: Screen shot of our app's virtual environment. Balls appearing at random locations determine the user's movements. On the left, the user moves upwards towards the green ball. On the right, a pink ball appears at a random spot and prompts the user to move towards its direction. Movements within the virtual space are effected by nudging the VR pointer towards a given direction.

Figure 3: A user wearing the head mounted display and using VR application in our study.

4.2 App Implementation and Data Collection

Unity 3D editor with Google VR plugin was used for designing the VR application. The application ran on a Samsung Galaxy S5 android phone which was mounted in the Google Cardboard VR device. At each time unit, the app logged the accelerometer and gyroscope reading, and also recorded the time when a user looked at a given ball. The sampling rate was 30 readings per second.

We collected data from 23 different subjects who were a mix of undergraduate and graduate students from our university. 17 of these were male while 6 were female. Each subject had two sessions that were at least a day apart. Subjects were allowed to preview the app before starting the experiment. In the experiment itself, subjects sat on a chair as they traversed the virtual space. Figure 3 shows an example of a participant undertaking our experiment. In each session, subjects repeated the experiment 15 times, spending a total of between 15 and 20 minutes.

4.3 Feature Processing

For each of the accelerometer and gyroscope, raw data from the VR app comprises of readings mapping to the x, y and z axes. From these values at each time stamp, we computed the magnitude feature, $m = \sqrt{(x^2 + y^2 + z^2)}$ so each sensor now had 4 readings at each time stamp. Given this data, we first undertook a smoothing step for each sensor's x, y, z and m streams. Smoothing was done using a low-pass, 10^{th} order Butterworth filter with a 5 Hz cut-off frequency. We then split the data in each stream into 12-second windows with a 50% overlap with the next window and proceeded to the feature extraction step which is described next.

Stream-level features: For each of the x, y, z and m streams in a window, we extracted the following two kinds of features: time domain and frequency domain features. These features were computed for both the four gyroscope streams and the four accelerometer streams. Table 1 shows the list of these features.

Time Domain Features	Frequency Domain Features
Range	Spectral Flatness
Interquartile Range	Spectral Skewness
Third Moment	Spectral Kurtosis
Fourth Moment	Spectral Centroid
Variance	Spectral Spread
Absolute Sum	Spectral Rolloff
Root Mean Square	Spectral Entropy
Mean	Energy
Skewness	
Kurtosis	
25th Percentile	
75th Percentile	

Table 1: Time domain and frequency domain features computed per window for each sensor and each of the x, y, z and m streams.

Cross-stream features: Beyond the features computed for each stream, we also computed a set of features that cut across multiple streams. These features include cross correlations between streams, norms of matrices created by concatenating multi-stream readings into a single matrix, and rotation features such as the pitch, roll and yaw. Table 2 shows a list of these features. To compute the matrix norms (i.e., 1-norm, infinity norm and frobenius norm), we concatenated the x, y and z streams of each sensor into a matrix whose norms we then computed. The correlations, pitch, yaw and roll were computed using well-known formulae and we hence do not discuss details of the formulations.

Accelerometer & Gyroscope Features	Gyroscope Features
1-Norm	Average Pitch
Infinity Norm	Average Roll
Frobenius Norm	Average Yaw
Correlation YX	Standard Deviation Pitch
Correlation ZX	Standard Deviation Roll
Correlation ZY	Standard Deviation Yaw

Table 2: List of features extracted across multiple streams. A stream refers to the x, y, z or m component.

4.4 Classification Framework

The feature extraction process described in the previous section produced a total of 178 features across both sensors. We normalized these features using the min-max scheme with a minimum of 0 and maximum of 1, and then applied Principal Components Analysis (PCA) to cut down on dimensionality given that users had limited sample sizes. For the ensuing results section, all findings are based on features produced when 95% of the variance was retained after the PCA process.

With features extracted, classification was done using the python scikit-learn framework. To gain insights into how different classification approaches might impact performance, we performed classification using the Logistic Regression with the default parameters, and the Support Vector Machine (SVM) with polynomial kernel and C value of 100 because they gave us the lowest mean EERs.

During classification, the data from the first session was used for training while data from the second session was used for testing. For both training and testing, the ratio of genuine to impostor samples was 1 to 2. Instances of impostor samples were randomly selected.

5 PERFORMANCE EVALUATION

5.1 Overview of our Performance Evaluation Approach

Given a user traversing the virtual space, it is quite likely that the user will every once in a while execute body movements that are very different from his (or her) typical body movement signature. If the system is configured to reject the user after a single uncharacteristic body movement, such a system would very likely have a very high false reject rate and very low usability ratings. In the same vein, an impostor trying to use the system will occasionally have a few body movements matching those of the user. If the system accepts users based on a single correct body movement, chances are that the false accept rate will be high.

The above described problem is in general seen for many behavioral biometrics-based continuous authentication mechanisms given the high intra-user variability and low inter-user variability exhibited by these modalities. To overcome this problem and customize our performance evaluation for a continuous authentication setting, we deviate from the traditional notion of a reject/accept as follows: we use five consecutive authentication windows to represent a single authentication attempt, and consider the attempt to be rejected if the majority of the five windows are rejected. Otherwise,

we consider the attempt as accepted. When data from a new window becomes available, the oldest window is discarded, meaning that the next authentication decision used four of the previous five windows (i.e., a sliding window mechanism with a step size of one). We use a value of five since it gave us the best performance during preliminary test experiments.

One potential weakness of our approach is that authentication decisions cannot be rendered until a user has generated enough data to cover five windows. However given the incentive of minimizing the false-accept/false-reject problem described at the start of this subsection, this should not be a big problem especially if the windows are short. Moreover, once the first five windows are realized, the rest of the process goes on seamlessly.

Having modified the traditional notion of what counts as a rejection or an acceptance, we used the standard error metrics for the rest of our performance evaluation. In particular, we report results based on the Equal Error Rate (EER) — the error rate at which the probability of false acceptance is equal to the probability of false rejection.

5.2 Authentication System Error Rates

Table 3 shows the mean EER and standard deviation of EERs across the population. These values were computed based on a window size of 12 seconds, and PCA features retaining 95% of the variance exhibited by the original feature-set. The EER results reveal that the Logistic Regression classifier marginally outperformed the SVM, although the SVM demonstrated a much lower variance in the error rates across the population.

Classifier	Mean EER	Std EER
Logistic Regression	0.0739	0.0179
Support Vector Machine	0.1001	0.0051

Table 3: Mean and standard deviation of the EERs across the population. The EER is measured on a scale running from 0 to 1.

Figure 4 gives a more revealing look at the EERs, divulging information about how individual users performed. The figure shows that for both classifiers, approximately 30% of the population had an EER of zero while about 50% of the users had an EER less than or equal to 0.03%. The figure also shows that less than 10% of the population had EERs greater than 0.2%. These numbers point to the promise depicted by the body movement biometric in VR, as only a small number of users disproportionately contribute to the EER. The results suggest that some sort of failure-to-enroll policy that bars a small subset of poor performing users from enrollment might tremendously improve mean authentication error rates across the population.

Figure 4 further shows that the Logistic Regression classifier had about 20% more users under 10% EER than the SVM classifier. A more systematic exploration of the reasons behind the better performance seen by the Logistic Regression classifier is part of our future work.

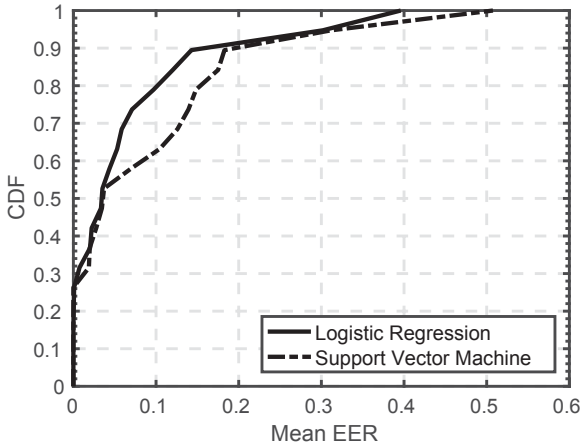

Figure 4: CDF of the EERs obtained across the population.

5.3 Sensitivity Analysis of Authentication System Parameters

To better understand how design parameters of the authentication system affected the EERs, we carried out a sensitivity analysis and in particular studied the impact of the window size from which feature vectors are derived, the amount of data used for training, and the percentage of variance retained during PCA. We briefly describe findings from this sensitivity analysis next.

5.3.1 Dependence of EER on window size. The window size determines the amount of data used to compute a single feature vector, and hence has a direct bearing on whether the features realistically capture a user's behavior. Very short window sizes enable faster feature computation and a short time to authentication, however, they are prone to noise since a very short burst of data could entirely be noise. Longer window sizes are in general more likely to capture more of the user's true behavior, however, they could tremendously slow down feature computation and by extension slow down the authentication process.

Figure 5 shows that as the window size increases, the EER decreases. This trend is in agreement with the expected behavior described in the previous paragraph. The figure further shows that the EER becomes relatively stable at window size of 12 seconds and above. This trend suggests that data longer than 12 seconds always captured enough of the user's behavior to offset any noise captured in a window.

5.3.2 Dependence of EER on training size. We varied the training dataset size between 20 and 30 instances per genuine user to understand how much of a user's own data is needed for adequate training of classifiers. As we varied the training data size, we kept the ratio of genuine to impostor samples fixed as 1:2 to avoid biasing the results due to variations in proportions of impostors and genuine samples across runs. The lower bound of 20 samples for this evaluation was chosen because values lower than 20 had very poor performance. On the other hand an upper bound of 30 was

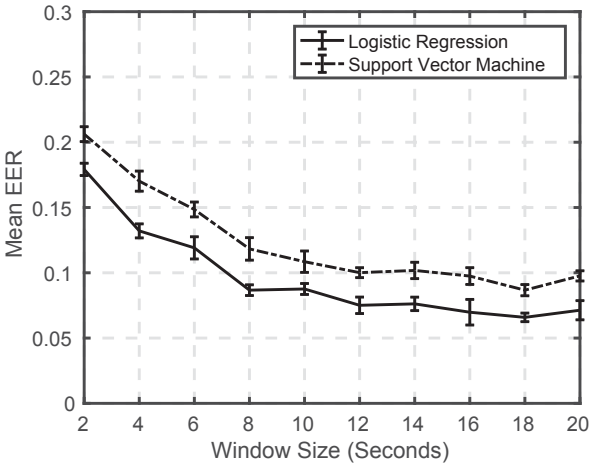

Figure 5: Mean EERs with different window sizes. The error bars represent a single standard deviation.

set because certain users begun running out data when numbers of samples exceeded 30.

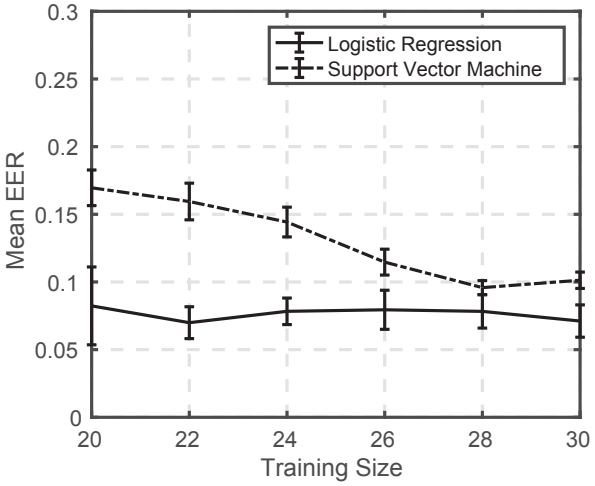

Figure 6: Mean EERs with different training data sizes. The error bars represent a single standard deviation.

Figure 6 shows that as we increase the training dataset size, the EERs reduce as expected. The figure also shows that starting from a training size of about 26 genuine instances, the EER begins to somewhat stabilize. It is noteworthy however, that in a practical VR application, the training size could be increased as much as possible given the potential availability of lots of genuine training data after the subject uses the app in question for a period of time. One of the continuing questions for our research therefore is how much lower the error rates could plummet if we had hours of genuine user data.

5.3.3 Dependence of EER on PCA percentage of variance retained. In the last segment of our sensitivity analysis, we studied how the percentage of variance retained after PCA affects classification performance. Retaining a smaller percentage of variance results into fewer components retained (or derived features) and hence minimizes both computations and storage. Retaining a higher percentage of variance means that most of the information in the original dataset is retained, however, this might not solve the dimensionality problems if a large number of components ends up being retained. In practice, for resource-constrained gadgets such as VR headsets, one would want to use the lowest percentage of variance giving acceptable classification performance.

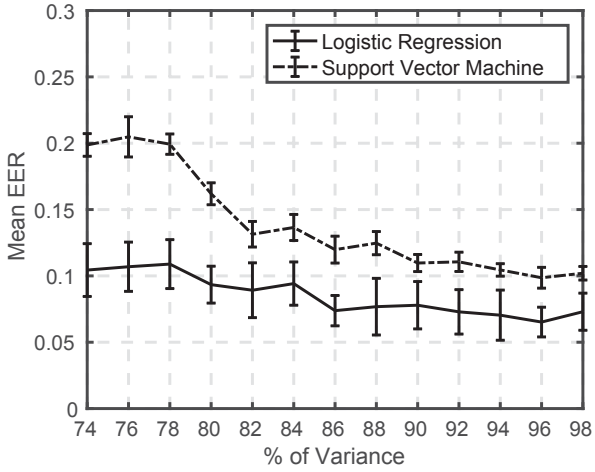

Figure 7: Dependence of mean EER on the percentage of variance retained during PCA. The error bars represent a single standard deviation.

Figure 7 shows that the EERs reduce as we increase the percentage of variance. The EERs stabilize at a percentage of variance of about 90%. This corresponds to a total of 70 components, a small fraction of the original set of 178 features extracted from the raw data.

6 CONCLUSIONS

In this paper we have designed and evaluated a head and body movement-based continuous authentication system for VR applications. Based on a dataset of 23 users, we have rigorously evaluated the performance of the system and carried out a sensitivity analysis on a number of system design variables, including the PCA settings, amount of training data and the window sizes used for feature computation. In our best configuration, we found the system to give EERs as low as 7%. These results indicate that the head and body movement-based biometrics holds promise and might provide answers to the looming VR authentication problem.

It is noteworthy that in practice our method would have to be app-specific since body movement patterns might vary across different apps. For example, a banking app would have to train on data collected from the same app just like a gaming app would have

to be trained on data collected from the same gaming app. This kind of app-centric learning is already widely used (e.g., health tracking mobile apps) and would be practical for VR as well. An interesting research question that is part of our ongoing work is that of how different VR apps apply to our method. Additionally, we are studying the design of alternative classification frameworks (e.g., deep learning) that might improve error rates.

ACKNOWLEDGMENTS

This research was supported by National Science Foundation Award Number: 1527795.

REFERENCES

[1] Leif P Berg and Judy M Vance. doi:10.1007/s10055-016-0293-9, 2017. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality* 21, 1 (doi:10.1007/s10055-016-0293-9, 2017).

[2] Magesh Chandramouli and Justin Heffron. 2015. A Desktop VR-based HCI framework for programming instruction. In *Integrated STEM Education Conference (ISEC), 2015 IEEE*. IEEE, 129–134.

[3] Ying-Chieh Chen, Jing-Jang Hwang, Ronggong Song, G. Yee, and L. Korba. 2005. Online gaming cheating and security issue. In *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, Vol. 1. 518–523 Vol. 1. https://doi.org/10.1109/ITCC.2005.215

[4] CNN. 2016. Alibaba offers VR shopping. Video. (5 Nov. 2016). Retrieved September 10, 2017 from http://www.cnn.com/videos/world/2016/11/28/alibaba-vr-shopping-stevens-pkg.cnn

[5] Simon Eberz, Kasper Bonne Rasmussen, Vincent Lenders, and Ivan Martinovic. 2015. Preventing Lunchtime Attacks: Fighting Insider Threats With Eye Movement Biometrics. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*. The Internet Society.

[6] Hui Fang, Jie1 Zhang, Murat2 ensoy, and Nadia1 Magnenat-Thalmann. 2014. Reputation mechanism for e-commerce in virtual reality environments. *Electronic Commerce Research and Applications* 13, 6 (2014), 409–422.

[7] Mohammed E Fathy, Vishal M Patel, and Rama Chellappa. 2015. Face-based active authentication on mobile devices. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 1687–1691.

[8] 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.

[9] Blizzard Games. 2016. Being accused of impersonating someone else. (5 July 2016). Retrieved September 7, 2017 from https://eu.battle.net/forums/en/overwatch/topic/17612302996

[10] Gearbrain. 2017. How VR and AR will change the way you shop. (17 March 2017). Retrieved September 7, 2017 from https://www.gearbrain.com/vr-shopping-apps-2312243800.html

[11] Walter J Greenleaf. 2016. How VR technology will transform healthcare.. In *SIGGRAPH VR Village*. 5–1.

[12] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. 2016. Demo of Headbanger: Authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 1–3. https://doi.org/10.1109/PERCOMW.2016.7457076

[13] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. 2016. Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–9. https://doi.org/10.1109/PERCOM.2016.7456514

[14] Hong Lu, Jonathan Huang, Tanwistha Saha, and Lama Nachman. 2014. Unobtrusive gait verification for mobile phones. In *Proceedings of the 2014 ACM international symposium on wearable computers*. ACM, 91–98.

[15] Upal Mahbub, Vishal M Patel, Deepak Chandra, Brandon Barbello, and Rama Chellappa. 2016. Partial face detection for continuous authentication. In *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2991–2995.

[16] Upal Mahbub, Sayantan Sarkar, Vishal M Patel, and Rama Chellappa. 2016. Active user authentication for smartphones: A challenge data set and benchmark results. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 1–8.

[17] Arik Messerman, Tarik Mustafić, Seyit Ahmet Camtepe, and Sahin Albayrak. 2011. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. In *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 1–8.

[18] Kenrick Mock, Bogdan Hoanca, Justin Weaver, and Mikal Milton. 2012. Real-time continuous iris recognition for authentication using an eye tracker. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 1007–1009.

[19] Isao Nakanishi, Sadanao Baba, and Chisei Miyamoto. 2009. EEG based biometric authentication using new spectral features. In *Intelligent Signal Processing and Communication Systems, 2009. ISPACS 2009. International Symposium on*. IEEE, 651–654.

[20] Joseph Roth, Xiaoming Liu, and Dimitris Metaxas. 2014. On continuous user authentication via typing behavior. *IEEE Transactions on Image Processing* 23, 10 (2014), 4611–4624.

[21] S Sangani, J Fung, R Kizony, ST Koenig, and PL Weiss. 2013. Navigating and shopping in a complex virtual urban mall to evaluate cognitive functions. In *2013 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, 9–14.

[22] Abdul Serwadda, Vir V Phoha, Sujit Poudel, Leanne M Hirshfield, Danushka Bandara, Sarah E Bratt, and Mark R Costa. 2015. fnirs: A new modality for brain activity-based biometric authentication. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. IEEE, 1–7.

[23] Abdul Serwadda, Vir V Phoha, and Zibo Wang. 2013. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*. IEEE, 1–8.

[24] Diksha Shukla, Rajesh Kumar, Abdul Serwadda, and Vir V. Phoha. 2014. Beware, Your Hands Reveal Your Secrets!. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, USA, 904–917. https://doi.org/10.1145/2660267.2660360

[25] Mohammad Tamviruzzaman, Sheikh Iqbal Ahamed, Chowdhury Sharif Hasan, and Casey O'brien. 2009. ePet: when cellular phone learns to recognize its owner. In *Proceedings of the 2nd ACM workshop on Assurable and usable security configuration*. ACM, 13–18.

[26] International Business Times. 2017. Amazon to deliver VR shopping experiences for next-generation retail therapy. (8 Feb. 2017). Retrieved September 7, 2017 from https://goo.gl/dT6LE2

[27] Guixin Ye, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. 2017. Cracking Android pattern lock in five attempts. In *The Network and Distributed System Security Symposium*.

[28] S. Yi, Z. Qin, E. Novak, Y. Yin, and Q. Li. 2016. GlassGesture: Exploring head gesture interface of smart glasses. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524542