GaitKeep: A Multi-Modal Approach to Gait Authentication

INTRODUCTION

In this paper we present GaitKeep, an exploratory method of mobile gait authentication. GaitKeep aims to accurately and robustly authenticate users via a multimodal gait analysis in a secure and usable manner. GaitKeep uses two types of sensors to authenticate users, utilizing a user's device and an external camera to create a biometric signature. By creating a hybrid-device approach to authentication, we hope to increase the resistance of our authentication system to potential attacks. Furthermore, we evaluate four possible authentication schemes for GaitKeep in order to assess performance and security: (1) mobile-only authentication, (2) visual-only authentication, (3) mobile-visual sensor fusion recognition, and (4) a mobile-visual recognition consensus.

BACKGROUND

Biometric authentication

Biometric systems use an individual's unique physiological or behavioural characteristics to identify or authenticate. Some examples of biometrics include fingerprints, facial features, keystroke patterns, or gait patterns. These biometric systems fall into two categories: verification and identification [1]. In verification systems, the individual claims an identity and presents a biometric signature; the system accepts or rejects the individual based on whether the claim is true, i.e. whether the identity matches the biometric signature. In identification systems, the system compares the individual's biometric signature with those of known users in a database, looking for a match. Performance of an identification system is measured by the percentage of queries in which the correct answer is found in the top few matches [1]. Performance of a verification system involves quantifying false rejects (system rejects a valid identity) and false-alarm (incorrectly accepts a false identity) [1] which can be reframed as false negatives and false positives.

Gait analysis and recognition

Gait analysis is the process of systematically studying an individual's movement patterns. Gait analysis is typically accomplished using data from videos, floor sensors, or wearable sensors (i.e., accelerometer or gyroscope) [2]. Gait recognition refers to the process of identifying individuals through an individual's walking pattern. Recognition through vision, where gait features are extracted via image and video processing techniques, has seen success in large surveillance and information systems. Similarly, worn sensor gait recognitions have achieved similar success via deep learning techniques. Gait as a biometric offers high usability for continuous and unobtrusive identification [14].

Security of biometric systems

Eberz et. al., [6] evaluate the susceptibility of different biometric authentication systems to "cross-context" attacks. In such attacks, the adversary may obtain biometric data from an insecure source context. Subsequently, the victim may be impersonated in a target context. For example, an accelerometer-based gait authentication system may be susceptible to an attack where the user's gait data is taken from a fitness tracker or mobile fitness app. In the face of such attacks, gait was found to be twice as unpredictable compared to others such as touch dynamics, ECG, eye movements, and mouse movements [6]. The high unpredictability of gait as a biometric is indicative that it is a robust form of authentication regardless of its specific implementation [6].

In considering the security of our own system, Eberz et. al., [6] suggests that we should adjust false accept and false reject thresholds accordingly with the vulnerability of our sensor modalities, and that the biometric template should be stored in a trusted module inaccessible to the adversary. Furthermore, we must consider the on-body location of the accelerometer sensor and the potential for adversaries to collect gait data across contexts. These considerations help justify the additional use of an external camera for gait recognition.

Mobile Sensing

RAPID [4] has shown a device-free, multi-modal node approach to recognition in the wild. Their approach uses both WiFi and acoustic information from a user's footstep. It identified subjects with an average accuracy of 86% to 92% depending on the size of the test groups. These results provided inspiration for us to develop a (hybrid) device-free approach to recognizing gait, as RAPID showed this approach yields increased accuracy. An alternate implementation choice was inspired by CenceMe [5]. The results of CenceMe suggested that our app could preprocess the data by taking descriptive statistics of the accelerometer and gyroscope values. However, our implementation could not take this approach because gait-recognition cannot be (robustly) done with just descriptive statistics, and we needed the raw data to uniquely classify individuals. In order to use high-accuracy recognition systems, we needed the complete data (as opposed to reduced versions with descriptive statistics) in order to classify. Data issues and the difficulty of porting over models on mobile has made us decide to upload and classify on a centralized server.

OUR SOLUTION

Our contribution is a method of gait authentication that takes a hybrid-device approach to authenticating users. Our aim is to build a system which identifies users accurately, securely, and in a usable manner. We do this by treating the user's device as a body-worn sensor and using an external camera for vision analysis. Data collected from these two sensor modalities is used to featurize individual gait patterns. The server will use this to authenticate the user.

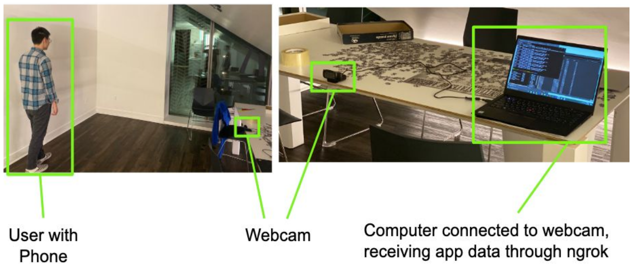

User with Phone
Webcam
Computer connected to webcam, receiving app data through ngrok

Figure 1. The user walks a distance of 2 meters with a phone in hand, or at the side of the body while a webcam records (left). The webcam is connected to the recording server (right).

System Architecture:

The architecture of our multimodal gait recognition system consists of a mobile device, webcam, and webserver. The mobile device and the webcam serve as sensors that measure accelerometer values and visual data of the authentication user.

The webserver is an abstraction that provides synchronization and automated labeling of mobile and visual data, in addition to being a resource to offload classification computation onto. It may not be necessary to implement a web server at all and alternative approaches are encouraged.

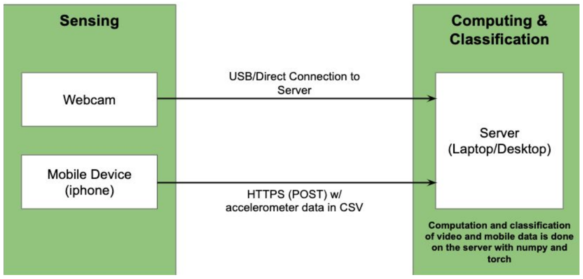

Sensing
Webcam
Mobile Device (iphone)
USB/Direct Connection to Server
HTTPS (POST) w/ accelerometer data in CSV
Computing & Classification
Server (Laptop/Desktop)
Computation and classification of video and mobile data is done on the server with numpy and torch

Figure 2. Abstracted system architecture: two sensing components connected to a server that is responsible for computation and recognition.

User recognition is not completed on the device but instead the same remote server that the app sends data to. This allows the app to use minimal resources and be power efficient. It also allows us to compare sensor data with visual data in a more elegant manner off the device.

User Application

We developed an iOS application for the iPhone 11 that collects accelerometer and gyroscope data from the phone's sensors. The application has two modes: data collection and data classification. The app communicates with the server through the HTTPS protocol. In data collection mode, the app asks the user what they would like to name the file which holds their sensor data and then records the user's walking data. The app formats the data in a CSV file format and sends it to our classification server using POST requests. The app also uses a GET request to start and stop a webcam that visually records the user as they walk while the phone's sensors are also recording their movement. Using this application, we create a seamless and simple way for a user to record their data in order to be used as an authentication method.

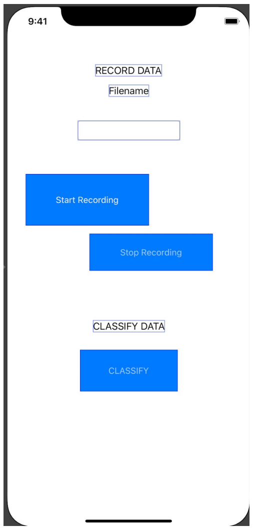


Figure 3. iOS Application and two modes. Data collection mode records data and sends it to a remote server. Data Recognition records data and returns if a user is identified.

The data recognition mode works similarly to the data collection mode, but also returns a result of the user identification to the user and a device to be unlocked. If the app is unable to identify the user then it returns that recognition failed. If the app does identify the user, then it returns that the user was recognized and if a user has linked a device, it will unlock the device.

Data Collection

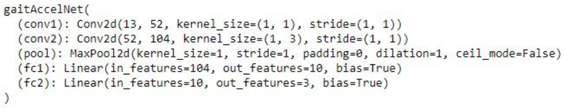
Visual and mobile data is collected simultaneously through the system architecture as described. Once the user presses “start” on the app, the server receives requests and begins the recording process.

The data collected consists of 30 traces of accelerometer and visual data for each member with a total of 90 traces. Going forward, having 2-3 non-members traces would be ideal, allowing us to more accurately assess the performance of our classifiers. The dataset for reproducibility can be found in the references [15]. In addition, the training and validation of the models was done on 80/20 split.

Convolutional Accelerometer Recognition

The module responsible for classification based solely on accelerometer values is GaitNet - a convolutional neural network. The network works by treating data traces with T entries for N features as a T x N matrix and convolving over this matrix with a kernel of different sizes. It then maxpools the output of the convolution layers into two fully-connected layers (like a traditional neural network). GaitNet uses stochasic gradient descent with a learning rate of 0.001 and a momentum of 0.05. GaitNet is a much shallower network and takes less features and shorter trace lengths compared to similar networks [12].

Since our dataset was an equal split between the authors of the paper, no weighting was associated with the cross-entropy loss; however, this cross entropy loss matters when creating randomized training/validation sets with the training of our GaitNet classifier. Note that the output of GaitNet will need to have a softmax transformation to turn a forward pass into probability distribution on the classifications.

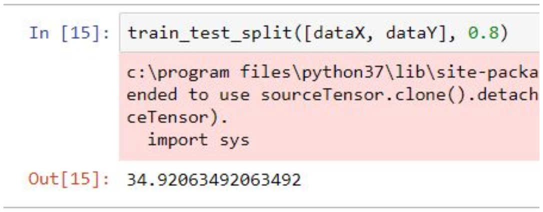


```
gaitAccelNet(
  (conv1): Conv2d(13, 52, kernel_size=(1, 1), stride=(1, 1))
  (conv2): Conv2d(52, 104, kernel_size=(1, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=1, stride=1, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=104, out_features=10, bias=True)
  (fc2): Linear(in_features=10, out_features=3, bias=True)
)
```

Figure 4. The layers of GaitNet (output from torch)

Something that may improve the performance of GaitNet is to use more features (like gyroscope in each axis) or taking longer traces of the users walking behavior.

On average, GaitNet has ~34-35% accuracy. This indicates that GaitNet has no strong predictive power (because a classifier that always guessed a single result would have a 33.3% accuracy rate. Interestingly, some results of our GaitNet prediction shows a 0% or 75% accuracy rate - which indicates that our classifier has discriminative potential (in the 0% accuracy) and predictive potential (in the 75% accuracy) - however, these results only occur 10% of the time - meaning that this observation is just by chance..



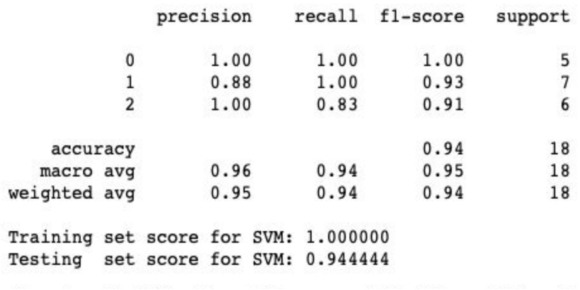
```
In [15]:  train_test_split([dataX, dataY], 0.8)

          c:\program files\python37\lib\site-packa
          ended to use sourceTensor.clone().detach
          ceTensor).
            import sys

Out[15]:  34.92063492063492
```

Figure 5. An example train/validation split that trains GaitNet and evaluates the performance on a random validation set. The output is accuracy (34.9%)

There are several possible factors for the poor performance of GaitNet relating to data-recording configuration, length of traces, and lack of additional features. The explanation behind the performance of GaitNet will be expanded on in the conclusion.

Visual Recognition

The module which performs visual recognition consists of a pretrained model accessible on GitHub [13], which takes raw RGB video frames as input and outputs an identification vector. The identification vectors for each user's gait are linearly separable and thus classifiable. Two subnetworks, HumanPoseNN and GaitNN [13], translate spatial features to pose descriptors, then temporal features are pooled into a one-dimensional identification vector. We train a multi-class SVM classifier to distinguish the identification vectors of each registered individual from the others. Hyperparameter tuning is done using scikit-learn's GridSearchCV, which stands for grid search cross validation. The SVC() model from sklearn is used to fit the model to our training data, a sequence of identification vectors and their labels in tuples. We pass in a grid of parameters, and SVC helps us find optimal parameters. Our best regularisation parameter C is 1, the best kernel used to transform data to a higher dimension is gaussian RBF, and the best gamma is 0.001.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         5
           1       0.88      1.00      0.93         7
           2       1.00      0.83      0.91         6

    accuracy                           0.94        18
   macro avg       0.96      0.94      0.95        18
weighted avg       0.95      0.94      0.94        18

Training set score for SVM: 1.000000
Testing  set score for SVM: 0.944444
```

Figure 6. Performance metrics of our vision recognition system for users 0-2.

EVALUATION

Authentication schemes

The performance of our system depends on what authentication scheme we use. The possible mechanisms we have considered are authentication via: (1) feature vectors generated by only accelerometer data, (2) feature vectors generated by visual data only, (3) concatenated accelerometer and visual feature vectors and (4) a consensus mechanism.

Evaluation of our system depends on whether we use it for identification or verification. If we seek to identify, such as in methods 1-3, the efficacy of our system is limited by model recognition performance. If we seek to verify, such as in method 4, we want to tune acceptance thresholds to minimize false rejects and false alarms.

CONCLUSION & DISCUSSION:

The first step to having a more complete version of GaitKeep is to implement and evaluate sensor fusion and recognition consensus mechanism. From the results of our work the vision-based gait recognition based on HumanPoseNN and GaitNN significantly outperform GaitNet. This means that the additional features from GaitNet may provide marginal or insignificant improvement if we were to fuse our raw data or extract output from intermediate layers of both networks. The worst case scenario is if the current iteration of our mobile accelerometer data interferes with authentication results. An improved version of GaitNet or changes to the data-collecting configuration may be necessary for real improvements.

We would like to collect more users (n > 3) so that the training and evaluation of our multimodal recognition systems is much richer (opposed to the 1/3 chance of guessing correctly).

In addition, we would want to calculate richer metrics such as false positives and false negative rates with more individuals to provide greater insight to the security of our system. These false positive and false negative rates can give us insights into the security and usability of GaitKeep, which are tied to the false positive and false negative rates respectively.

Without performing the sensor fusion or the recognition consensus mechanisms, the visual gait-recognition portion of Gait-Keep outperforms the mobile-data version.

Mobile recognition might also be improved by increasing the feature richness or length of our traces. In our initial iteration, we excluded gyroscopes due to implementation complications. Furthermore, research suggested that gyroscope data is unnecessary in gait recognition. However, other research exploring convolutional approaches to gait recognition use both accelerometer and gyroscope data, in addition to including longer traces [12]**.** With only three feature vectors and the shortness of our traces we may not be able to extract enough meaningful features from convolutions - which could be the reason behind the poor recognition.

Furthermore, an analysis of our traces and video files show that the user might naturally stabilize the phone while walking with it – which is very different from other implementations that tie mobile devices to both multiple body parts or in pockets. A richer analysis would also try to look at the probability distribution outputs of our trained model and compare it to the dataset.

Finally, future work would include research and testing into various attacks on a more completed version of GaitKeep in order to evaluate the security viability of the system to basic attacks, and to tune the thresholds of each recognition modalities accordingly.

Group Member Contributions

April: Computer vision recognition, sensor fusion, and security

Ausar: Accelerometer data collection, mobile app development, and web communication with the server

Ted: Mobile-device gait authentication, video capture with sampling, and server functionality development

REFERENCES

[1] A. Jain, A. Ross, "An introduction to biometric recognition", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4-20, 2004.

[2] Gafurov, D.: A survey of biometric gait recognition: Approaches, security and challenges. In: NIK conference (2007)

[3] P. J. Phillips, A. Martin, C. L. Wilson and M. Przybocki, "An introduction evaluating biometric systems," in *Computer*, vol. 33, no. 2, pp. 56-63, Feb. 2000.

[4] Yuanying Chen, Wei Dong, Yi Gao, Xue Liu, and Tao Gu. 2017. Rapid: A Multimodal and Device-free Approach Using Noise Estimation for Robust Person Identification. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 3, Article 41 (September 2017), 27 pages.

[5] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. 2008. Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys '08). Association for Computing Machinery, New York, NY, USA, 337–350.

[6] S. Eberz, G. Lovisotto, A. Patanè, M. Kwiatkowska, V. Lenders and I. Martinovic, "When Your Fitness Tracker Betrays You: Quantifying the Predictability of Biometric Features Across Contexts," *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, 2018, pp. 889-905.

[7] Huan Feng, Kassem Fawaz, and Kang G. Shin. 2017. Continuous Authentication for Voice Assistants. In Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17). Association for Computing Machinery, New York, NY, USA, 343–355.

[8] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. 2010. MAUI: making smartphones last longer with code offload. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys ’10). Association for Computing Machinery, New York, NY, USA, 49–62.

[9] Tiffany Yu-Han Chen, Hari Balakrishnan, Lenin Ravindranath, and Paramvir Bahl. 2016. GLIMPSE: Continuous, Real-Time Object Recognition on Mobile Devices. GetMobile: Mobile Comp. and Comm. 20, 1 (July 2016), 26–29.

[10] M. P. Mufandaidza, T. D. Ramotsoela and G. P. Hancke, "Continuous User Authentication in Smartphones Using Gait Analysis," IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 4656-4661.

[11] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, " Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition", IPSJ Trans. on Computer Vision and Applications, Vol. 10, No. 4, pp. 1-14, Feb. 2018

[12] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, Q. Li, “Deep Learning Based Gait Recognition Using Smartphones in the Wild”, arXiv:1811.00338v2

[13] https://github.com/marian-margeta/gait-recognition

[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 (ISWC ’14). Association for Computing Machinery, New York, NY, USA, 91–98. DOI:https://doi.org/10.1145/2634317.2642868

[15] https://drive.google.com/open?id=1jz4an8mOoekfOSi-VQ2m11xnz1L-pE6Q