On the Feasibility of Handwritten Signature Authentication Using PPG Sensor

A B M Mohaimenur Rahman*, Yetong Cao†, Xinliang Wei‡, Pu Wang*, Fan Li†, Yu Wang‡
* Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina, USA
† School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
‡ Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania, USA

2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC) | 978-1-6654-3161-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCNC49033.2022.9700557

Abstract—**Handwritten signature authentication is an important service to defend against fraudulent activities. Current automated solutions rely heavily on dedicated devices and require certain user efforts. In this work, we explore the feasibility of a new type of signature authentication system, SAP - *Signature Authentication with PPG Sensor*, which leverages Photoplethysmography (PPG) sensors in wrist-worn wearable devices. To make SAP non-intrusive and secure, we design effective algorithms to separate the signature signals from the heartbeat signals in the raw PPG signals. We implement a low-cost hardware prototype of SAP. Our preliminary experimental results show that SAP can achieve an average F1 score of up to 98%.**

I. Introduction

The *signature authentication problem* studied here is primarily a user verification problem over a handwriting signature via wearable sensing data. As any other authentication systems, a user will give sensing data of his/her signature to claim to be a certain person. The system should verify the claim whether the user is a legit or illegitimate user. This is useful in many applications to prevent fraudulent activities. Different automatic handwritten signature authentication systems [1] have been developed, including wearable device based solutions (e.g., using motion sensors [2], [3] and acoustic sensors [4]). While these methods enjoy the low cost, non-intrusiveness, and easy deployment, many of them still require users' extra effort (for calibration or training) and suffer from low accuracy (due to environmental noises or attacks).

In this work, we investigate the feasibility of a signature authentication system to verify user's handwritten signatures by leveraging the Photoplethysmographic (PPG) sensor data from a wrist-worn wearable. As shown in Fig. 1, the system is trained on that registered signatures to create a model based on the features extracted from those signatures. When a new signature input is given, the system based on the saved models decides whether the user is a legitimate user or an attacker. Particularly, we exploit PPG sensor from wearable devices to capture the dynamic information of the blood volume change under the skin when a user is writing his/her signature. Our solution is different from existing PPG-based user authentication since a handwritten signature is beyond traditional user authentication. We more focus on the dynamic part of PPG-signal caused by the hand/finger movements. We have explored several segmentation algorithms to separate the signature signals from the raw PPG signal consisting of both heartbeat and signature signals. We also build a low-cost hardware prototype of SAP using a commercially off-the-shelf PPG sensor and a micro-controller to evaluate the feasibility of the proposed system. The experimental results show that SAP can achieve an average F1 score of up to 98%.

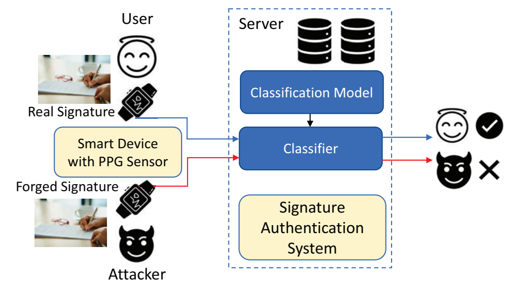


Fig. 1. Handwritten signature authentication via PPG sensor.

II. Challenges and Overall Design

To successfully validate the fine-grained signatures via the captured PPG signals from a wrist-worn device, there are primarily three challenges to be addressed. (1) *Coarse-grained Wrist PPG Signals:* PPG signals are relatively coarse-grained, noisier, and interfere with other signals. To tackle this, we design noise filtering to extract critical landmarks and apply feature extraction to extract discriminating features for fine-grained authentication. (2) *Same user having different PPG readings:* The PPG readings are different even for the same user, as the pulse signal may vary due to the effect of pressure and emotions of the user. It is challenging to use the PPG signals for signature authentication directly. To address this, we extract the signature portion signal from the whole signal so that the effect of pressure and emotion is minimized. (3) *Effect of the placement of the PPG sensor:* The sensor placement can also affect the performance since PPG readings vary at different locations. Via experiments, we find out that the readings at *median antebrachial vein* placement would represent the significant motion artifacts occurring due to the signature writing. Though this position is not in line with most people's wearing habits, it is feasible because the user will just have to rotate the wearable device while writing the signature.

Fig. 2 shows the overall architecture of SAP, which consists of four parts: *Data Collection*, *Data Processing and Segmentation*, *Feature Characterization*, and *Classification*.

In *data collection* module, the wrist-worn wearable device with a PPG sensor is used to collect the PPG data while the user is writing the signature on a sheet of paper.

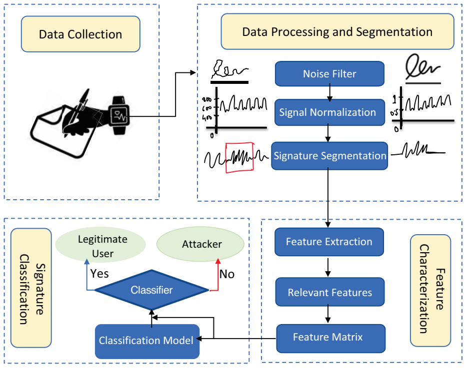


Fig. 2. System architecture of SAP.

Data processing and segmentation module has three steps: noise filter, signal normalization, and signature segmentation. The PPG signals first pass noise filters (Savitzky-Golay (S-G) filter [5]) and are normalized, then the signature part of signals are segmented from the whole signals to remove the pulse part. We explore three different segmentation methods: (1) *Skewness-DTW* method, in which the start and ending index of the signature portion is detected based on the skewness values of the signal and by the Dynamic Time Warping (DTW) technique similar to [6]; (2) *Dynamic Programming* (DP); and (3) *Binary Search* (Binary) methods. DP and Binary are offline change point detection methods inspired by [7].

Feature characterization module includes feature extraction, relevant features, and feature matrix. This module mainly deals with the extraction of relevant intrinsic characteristics of the input signal data that can discriminate each user from another. We use a time-series feature extraction tool *tsfresh* [8] to generate more than 700 time-series features. Then using its feature selection, we select 8 features primarily as the most relevant features.

Finally, the *classification* module performs the authentication with a trained classifier. The classification model is trained on the whole dataset of a user to classify a new incoming signal into any of the two categories: legitimate user or attacker. We use the different standard classifiers: Random Forest (RF), SVM (RBF Kernel), Gradient Boosting (GB), k-NN (kNN), Multilayer Perceptron (MLP), Feed-Forward Neural Network (NN), and OnevsRest (OvR).

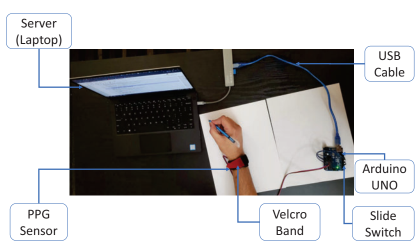
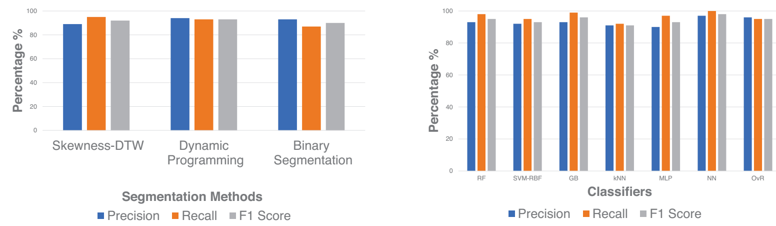


Fig. 3. SAP Prototype.

III. Prototype and Preliminary Evaluation

Although the commercially available smartwatches /fitness-trackers use PPG sensors to measure the heartbeat/pulse, they do not provide access to the raw PPG data. Thus, we build our own low-cost proof-of-concept PPG-band, as shown in Fig. 3 to imitate wrist-worn wearable devices to validate the feasibility of SAP. It consists of a velcro wristband, a PPG sensor by World Famous Electronics with a green LED, a USB cable, an Arduino UNO micro-controller, and a slide switch to start and stop collecting data. The PPG sensor is strapped to the Velcro band so that it remains facing towards the wrist when the Velcro band is worn. All software modules of SAP are implemented in the server using *Python* 3.8.3. We also use the *tsfresh* tool [8] for feature extraction, and the python package *scikit-learn* [9] for building the classifiers.



(a) segmentation methods (b) classifiers

Fig. 4. Performance of SAP for (a) different segmentation methods (average over all classifiers with 60% training data), (b) different classifiers (Skewness-DTW with 80% training data).

Because of the logistic limitations being created due to the COVID-19 pandemic situation, 5 healthy volunteers participated in the experiment. Participants take part in 6 sessions of signature writing. During each session, the participants provide 20 valid signatures, 10 random forgeries, and 10 skilled forgeries against a certain user who is considered as the legitimate user for the system providing 20 valid signatures, and 10 invalid signatures per session. We adopt the same definition of random and skilled forgeries from [2]. Totally, we collected $1,140$ samples of signatures. As the COVID-19 situation is relaxing now, we plan to collect more data to expand our findings further.

We perform tests on three signature segmentation algorithms and all seven classifiers. Fig. 4(a) shows the average performance of SAP with all the classifiers when the dataset was split into 60% training and 40% testing. Among the three segmentation methods, Skewness-DTW and DP both performed well. We then compare 7 commonly used classifiers under the Skewness-DTW segmentation and with a training size of 80%. The F1 scores for each classifier are shown in Fig. 4(b). Feed-Forward Neural Network performs the best with the highest F1-score of 98%, while the scores of RF/GB/OvR are also within a close range.

References

[1] D. Impedovo and G. Pirlo, "Automatic signature verification: The state of the art," *IEEE TSMC, Part C*, 38(5):609–635, 2008.
[2] A. Levy, B. Nassi, Y. Elovici, and E. Shmueli, "Handwritten signature verification using wrist-worn devices," *Proc. ACM IMWUT*, 2(3), 2018.
[3] G. Li and H. Sato, "Handwritten signature authentication using smartwatch motion sensors," in *Proc. of IEEE COMPSAC*, 2020.
[4] M. Chen, J. Lin, et al., "Silentsign: Device-free handwritten signature verification through acoustic sensing," in *Proc. of IEEE PerCom*, 2020.
[5] A. Savitzky, M.J.E. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analy. Chem.*, vol.36, p.1627, 1964.
[6] T. Zhao, J. Liu, et al., "PPG-based finger-level gesture recognition leveraging wearables," in *Proc. of IEEE INFOCOM*, 2018.
[7] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020.
[8] M. Christ, et al., "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, 05, 2018.
[9] F. Pedregosa, et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.