A SLIDING WINDOW BASED VOTING CLASSIFIER FOR ACTIVITY SENSOR BASED USER IDENTIFICATION

by

Sai Ram Vallam Sudhakar, BTech

THESIS
Presented to the Faculty of
The University of Houston-Clear Lake
In Partial Fulfillment
Of the Requirements
For the Degree

MASTER OF SCIENCE

in Computer Science

THE UNIVERSITY OF HOUSTON-CLEAR LAKE
MAY, 2021

A SLIDING WINDOW BASED VOTING CLASSIFIER FOR ACTIVITY SENSOR BASED USER IDENTIFICATION

by

Sai Ram Vallam Sudhakar

APPROVED BY

____
Kewei Sha, PhD, Chair

____
Wei Wei, PhD, Committee Member

____
Kwok-Bun Yue, PhD, Committee Member

RECEIVED/APPROVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

____
David Garrison, PhD, Associate Dean

____
Miguel A. Gonzalez, PhD, Dean

DEDICATION

I dedicate this thesis to my friends and my parents. Without their encouragement, understanding, and most of all love, the completion of this work would not have been possible.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my thesis advisor, Dr. Kewei Sha, for giving me the opportunity to do research and providing invaluable guidance throughout this research. I was highly motivated with his valuable comments. Without his support, I could not have finished this work. It was my pleasure working with him and I am proud to be a part of his research.

Beside my advisor, I would like to thank the members of my committee, Dr. Wei Wei and Dr. Kwok-Bun Yue for their precious time, support and suggestions in improving the quality of this thesis.

I would also like to thank Namrata Khayasta for collaborating on this research project and all her valuable comments on the project. My research work was continuation to her thesis. It was my pleasure working with her.

I would also like to thank all the reviewers from Elsevier journal who provided their insightful comments on our research project.

Finally, I want to thank my family for their unconditional love and support.

ABSTRACT

A SLIDING WINDOW BASED VOTING CLASSIFIER FOR ACTIVITY SENSOR BASED USER IDENTIFICATION

Sai Ram Vallam Sudhakar

University of Houston-Clear Lake, 2021

Thesis Chair: Kewei Sha, PhD

Identification is the core of any authentication protocol design as the purpose of the authentication is to verify the user's identity. The efficient establishment and verification of identity remain a big challenge. Recently, biometrics-based identification algorithms gained popularity as a means of identifying individuals using their unique biological characteristics. In this thesis, we propose a novel and efficient identification framework, ActID, which can identify a user based on his/her hand motion while walking. ActID not only selects a set of high-quality features based on Optimal Feature Evaluation and Selection and Correlation-based Feature Selection algorithms but also includes a novel sliding window based voting classifier. Therefore, it achieves several important design goals for gait authentication based on resource-constrained devices, including lightweight and real-time classification, high identification accuracy, a minimum number of sensors, and a minimum amount of data collected. Performance evaluation shows that ActID is cost-effective and easily deployable, selects only a minimum number of 10 high-quality features, uses only accelerometer sensor and increases the cost efficiency of user identification, collects only a small amount of 10 seconds of activity data, satisfies real-time requirements, and achieves a high identification accuracy of 100% when applied to a 30 user dataset.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Significance

User authentication is a valuable method for preventing unauthorized access to sensitive data. The object of authentication is to verify the identity of the user, so identification is an important part of the authentication protocol design [1, 2, 3, 4]. Several recognition technologies that can uniquely identify users and prevent impersonation have been established over the last few decades. It's critical that these recognition solutions provide a realistic and cost-effective method of rapidly identifying users while still offering a friendly user experience. In the modern world, username/password identity is commonly used [5], but it is vulnerable to hacking, theft, and fraud. Another common method for creating a verifiable identity is to use a digital signature based on cryptographic algorithms [6]. It's a successful solution, but it necessitates a powerful processor to produce digital signatures, making it impossible for devices with limited resources to create such an identity. Physical Unclonable Function (PUF) [7], a hardware-based solution, has recently emerged as a means of identifying users, and many authentication protocols are based on it. PUF is a powerful identity solution, but it necessitates additional hardware. Tokens and access cards [8] also have a hardware-based identification solution.

Biometric-based identity solutions are the next step in the identification and authentication process [9]. Because of the following factors, they are thought to be more successful than the previously listed digital identities. First and foremost, biometrics are an inherent part of the user's identity. Biometrics, unlike other standard methods of checking identity, such as usernames/passwords, PINs, tokens, and so on, cannot be forgotten, lost, or stolen [10]. Second, biometrics are one-of-a-kind for each individual, making them difficult to imitate. Third, biometric identities can be easily verified by analyzing biometric characteristics [11].

Many biometric identities have been created and implemented in modern computer systems. Face recognition techniques are used by the iPhone X and later models, as well as the Microsoft Surface Pro, to recognize legal users [12, 13]. In smartphones and computers, the fingerprint is the most commonly used biometrics-based identification [14]. Other common biometrics-based identities include ECG/EEG patterns [15, 16], Iris patterns [17], and palm vein patterns [18]. To capture biometrics, all of these solutions require specialized hardware. This can be costly, inconvenient, and intrusive to the user's experience.

Researchers studied human behavior patterns using data obtained by activity monitors such as accelerometers and gyroscopes and discovered unique characteristics that can be used as an identification. Activity sensors have been used in the literature to classify users based on keystroke dynamics [19], hand gestures [20, 21], and gait patterns [1, 2, 3, 4, 10, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. These current methods yield promising results, but the majority of them either employ computation-intensive algorithms based on a large number of features or use multiple sensors deployed across the body, which is not practical in real-life scenarios. The limitations of these approaches in the real-world applications include willingness to use wearable sensors, ability to wear them, success rate, scalability, ease of use, battery life, and the approach's usefulness [41]. Many sensors, such as accelerometers and gyroscopes, embedded in smartwatches and wristbands, can be used as biometric measuring devices as these devices become more widely accessible. As a result, using these sensors to measure biometrics in a cost-effective and convenient manner, we can design solutions that build and validate digital identity for users. In addition, this approach is not expensive and can be used in continuous authentication since it does not require any user interaction with the device.

1.2 Motivation and Research Challenges

The recent solutions of behavioral biometrics are inexpensive, more appropriate than conventional biometrics and/or they can be used in combination with traditional biometrics such as multi-factor authentication to improve security and usability [42]. Current biometrics-based authentication systems require user interaction, which is inconvenient for the user. Typing the password, lifting the phone for face id, and pressing the fingerprint sensor are just a few examples. This would be much more difficult for the user during continuous authentication, as the user must authenticate several times [43, 44, 45]. This problem can be solved by activity sensor based identity solutions such as wearable sensor based gait recognition [46], touch gestures based recognition [47], keystroke based recognition, etc., since the biometric patterns are captured implicitly while the user interacts with the device [48]. These approaches address the privacy [41] and power consumption [42] issues better than traditional vision based activity recognition.

Many previous approaches collected activity data by installing sensors on various parts of the body, which is inconvenient for the user. The ideal position for sensors, we assume, is on the user's wrist. To keep the design cost-effective, only a few sensors must be used, and existing smartwatch sensors must be used. To minimize time complexity, only a small amount of data must be obtained, classification algorithms must be lightweight, and at the same time accuracy must be high.

1.3 Novelty of the Research

We propose ActID, an efficient system for activity sensor-based user identification, in this study, to efficiently identify users using wrist-worn sensors. Our classification algorithm's main task is to overcome challenges posed by the authentication application's specifications and the application's resource-constrained devices. We want to create an efficient system that can reliably classify users in real time using a small number of sensors, a small amount of data, and only lightweight classification algorithms. Our proposed approach is unique in four ways. First, we evaluate the extracted features and select a set of high-quality features that can clearly identify individuals using the Optimal Feature Evaluation and Selection method (OFES) [49, 50, 51] and Correlation-based Feature Subset Selection (CFSS) [52] algorithms. As a consequence, the feature set can be kept to a minimum. It also helps to simplify the algorithm. Second, we introduce a novel classification algorithm for gait authentication called Sliding Window based Voting classifier [53], which reuses data to minimize data size and adapts voting to improve accuracy. Third, our proposed architecture provides a smooth user interface. Unlike other research approaches that require participants to wear several sensors on various parts of their bodies, our experiment only requires participants to wear one wrist sensor and walk normally on a flat surface for less than a minute to train the classifier. Fourth, by using only an accelerometer sensor, we reduce the number of sensors needed and increase the cost efficiency of user classification based on activity sensor data. The proposed system can achieve high accuracy of 100% when applied to a 30 user dataset based on a simple prototype with a multi-class classifier, which is better than similar methods such as those presented in [3, 4, 22, 26, 28, 32, 37, 39, 46].

1.4 Contribution

The study makes three distinct contributions. First, we looked at the difficulties of identifying users using activity sensors. Second, we improved the feature selection method of Kayastha et al. [50] by using correlation filter, sensor reduction method and compared the results with other similar approaches. Third, we proposed a new classification algorithm that incorporates a sliding window technique as well as a voting system. Finally, we built a prototype for activity sensor-based user recognition and compared with similar approaches.

1.5 Organization of Thesis

The remainder of the study is written out in the following manner. A group of similar works is mentioned in Section 2. The aim of this study is discussed in Section 3. The architecture of the ActID system is discussed in Section 4. The performance assessment is presented in Section 5 using a simple prototype implementation. In Section 6, we sum up the thesis and look forward to future work.

CHAPTER 2

RELATED WORK

Activity-based user identification has been a fascinating research subject. This section contains a list of works that are relevant to our study.

2.1 Activity Sensor-Based User Identification

Biometrics-based user identification is a good way to classify or validate people based on their physiological or behavioral traits [54]. Physiological biometrics is concerned with an individual's exact dimensions, measurements, and physical characteristics. Behavioral biometrics, in contrast to physical biometrics, can be easily obtained using existing hardware or wearable sensors that use less electricity, with only software needed for analysis. As a result, behavioral biometrics are both inexpensive and simple to use. Behavioral biometrics is the field in which our research falls.

Activity sensor-based user identification has demonstrated tremendous research promise in the field of behavioral biometrics in recent years. Gait is one of the most common activity-based biometric characteristics because it has been proven to be a feasible authentication method by researchers. Table 2.1 summarizes several recent findings on activity sensor-based gait recognition. Sensor-based gait authentication was first proposed by Ailisto et al. [1]. The acceleration sensor attached to the user's waist acted as the basis for their gait authentication. Cross-correlation was used as a test of similarity, and they got 6.4 percent EER. Gafurov et al. [3] expanded on their methodology and studied it. For gait authentication, some designs have used sensors attached to various body parts (e.g., leg, waist, hip, arm, and all over the body) [22], which is not realistic in real-life scenarios. As a result, these innovations have yet to be deployed on a large scale.

Study	**Subjects**	**Sensor Location**	**Results**
Ailisto et al. [1]	36	Waist	EER: 6.4%
Mantyjarvi et al. [2]	36	Waist	EER: 7% - 19%
Gafurov et al. [3]	21	Lower leg	EER: 5%, 9%
Al Kork et al. [22]	50	Leg, hand, wrist, pant pocket, shirt pocket and bag (left and right side)	EER: 0.17% - 2.27%
	23	Hand (holding smartphone)	EER: 1.23% - 4.07%
Derawi et al. [10]	51	Pocket attached to the belt (right-hand side of the hip)	EER: 20.1%
Rong et al. [24]	21	Waist	EER: 5.6%, 21.1%
Sun et al. [25]	22	ankle	EER: 3.03%
Kwapisz et al. [26]	36	Front pants leg pocket	Accuracy: 82.1%, 92.9%
Thang et al. [27]	11	Trouser pocket position	Accuracy: 92.7% (SVM)
Johnston et al. [28]	59	Waist (smartwatch)	EER: 2.6% - 8.1%
Kumar et al. [46]	12	Wrist (smartwatch)	Accuracy: 95%
Liu et al. [32]	7	Four positions, left wrist, chest, left ankle and waist	Accuracy: 86.7%

Table 2.1: *A summary of user identification based on activity sensor.*

2.2 Smartphone and wrist sensor-based user identification

Modern smartphones and wristwatches are fitted with powerful sensors that record activity sensor data from the people who use them. These devices have evolved into a rich data base for measuring human behaviors like walking, jogging, sitting, ascending stairs, and so on [29]. In comparison to other applications, these devices are unobtrusive, easier to transport, and easier to capture activity data for user identification. Table 2.2 summarizes several recent findings on smartphone and wrist sensor-based gait recognition. Nickel et al. [4] used the K-Nearest Neighborhood algorithm to create a method for extracting gait features and demonstrated its feasibility on smartphones, achieving an EER of 8.24 percent. Al Kork et al. [22] used wearable sensors and a smartphone to build a multi-model biometric database for human gait. They were able to reach an EER of 0.17 percent to 2.27 percent. At the same time, they used five sensor nodes on various body positions, as well as a smartphone with built-in accelerometer and gyroscope sensors that they kept in their hands. In contrast, we only used a single sensor node in our method. Furthermore, their data collection time is 4.5 minutes, while ours is 60 seconds, with just 10 seconds of data used. Garcia et al. [21] were the first to consider hand dynamics for authentication based on door opening movements. To collect sensor data, they used the accelerometer, gyroscope, and magnetometer installed in Google Nexus 4 smartphones. They suggested a machine learning-based method for classification that included a number of statistical and physical features as well as a Support Vector Machine (SVM). They were able to achieve a 92 percent accuracy rate using their system. Most experiments on smartphone-based gait recognition presume that the phone is in a fixed position (e.g., belt, pocket, or hand) such that differences in the walking pattern detected by motion sensors due to shifts in the phone's positioning (e.g., from pocket to hand) can be ignored [30]. However, in fact, there is no specific location of the phone on the user's body, and there is currently no proper system that can locate the phone's position automatically [46].

Study	**Subjects**	**Results**
Kumar et al. [46]	12	Accuracy: 95%
Primo et al. [30]	30	Accuracy: 82.3%
Liu et al. [32]	7	Accuracy: 86.7%
Johnston et al. [28]	59	Accuracy: 84%
Nickel et al. [4]	20	EER: 8.24%
Al Kork et al. [22]	23	EER: 1.23% to 4.07%
Johnston et al. [28]	59	EER: 2.6% - 8.1%
Garcia et al. [21]	20	Accuracy: 92%

Table 2.2: *A summary of smartphone and wrist sensor-based user identification.*

Since users typically wear their smartwatches or wristbands in the same position and orientation, wrist-wearables such as smartwatches and wristbands provide significant advantages over smartphones, especially in gait authentication. The wrist position offers more precise information about a user's movements than the most popular location for smartphones, such as pockets or handbags [28]. Wearable sensor-based movement recognition has a variety of uses in health care, patient or elderly tracking, recovery training, and a variety of other human interaction situations [31]. It is more realistic to collect activity data from wrist-wearables for user identification due to its rising popularity, position accuracy, and broad applicability.

Johnston et al. [28] used a smartwatch to gather gait data and found that features derived from accelerometer data had an EER of 2.6 percent and data derived from gyroscope data had an EER of 8.1 percent. They demonstrated their findings by training five minutes of the dataset with a maximum recognition accuracy of 84 percent using six types of features: average, standard deviation, average absolute difference, time between peaks, binned distribution, and average resultant acceleration. In each of the studies described above, a large amount of data was used to train the model. Our experiment needed just 10 seconds of data and yielded a 100% promising result.

Kumar et al. [46] suggested four continuous authentication designs based on arm movement characteristics as people walk. With the aid of a smartwatch's sensor, they were able to collect motion data. Their first design captures arm acceleration with an accelerometer sensor, their second design collects arm rotation with a gyroscope sensor, their third design uses a mixture of both accelerometer and rotation at the function stage, and their fourth design uses fusion at the score level.

Liu et al. [32] demonstrated a method for authenticating using 20 different features from the time and frequency domain in a recent report. In their proposed scheme, they used the C4.5 decision tree and obtained an accuracy of 86.7 percent. The author concluded that a feature selection strategy was needed to improve the model's performance and reduce computational complexity.

López-Fernández et al. [40] proposed a multi-view gait recognition on curved paths using local variations on the angular measurements along time. They have used a stream of images from a certain number of fixed cameras (Eg. surveillance cameras) to recognize a user based on gait patterns whereas we used activity data from the accelerometer sensor on the wrist of the user. We shared some similar ideas in the classifier design, but our design couples with a unique feature selection and our identification approach has fewer constraints on user movement and is less costly. In addition, we have also introduced a mechanism to obtain optimal values of the parameters such as window size and sliding interval during the sliding window process.

This study is based on but significantly extends Ms. Namrata Kayastha's Master's Thesis [50]. The differences are summarized below. The focus of the Kayastha's thesis is to develop a feature evaluation and selection mechanism, while in this study, we focus on designing a multi-class classification algorithm. In the evaluation of the Kayastha's thesis, the experiments are based on a 14-user single-session dataset, while in this study, the experiments are based on a 30-user two-session dataset. We have also improved the feature selection algorithms in this study by applying correlation analysis and sensor reduction. Consequently, the classification in this study is based only on a set of accelerometer data, while the classification in the Kayastha's thesis uses both accelerometer and gyroscope data. Furthermore, the Kayastha's thesis only utilizes and evaluates existing traditional classification algorithms, but we designed and evaluated a novel sliding-window-based voting classification algorithm in this study. As a result, the paper improves classification accuracy.

In summary, compared with many previous research, our experiment only uses 10 seconds of data and we tackle the challenges faced by the previous studies. With our proposed framework, we intend to keep the size of the feature set as small as possible, identify a set of high-quality features that can help distinctly identify individuals, provide a smooth user experience, as well as provide a promising result.

CHAPTER 3

MOTIVATION

One of the common problems of authentication is its intrusive way of authenticating users. It could be typing the password, raising the phone for face id, touching the fingerprint sensor, giving the voice to identify the user. These kinds of authentications require user interaction every time the user attempts authentication, which leads to lots of inconvenience to the user. Syed Zulkarnain et al. studied on finding the profile of an individual like age, gender etc., based on their behavioral characteristics of keystroke dynamics [43]. It would be interesting to see if the same can be achieved with activity sensor data. Heather Crawford et al. proposed a framework that integrates multiple behavioral biometrics to implement an effortless and continuous authentication mechanism without user interaction [45]. Similarly, Saevanee et al. proposed a novel text-based multimodal biometric approach using linguistic analysis, keystroke dynamics and behavioral profiling so that the number of intrusive authentication requests required for high security applications will be decreased [44]. All these approaches are trying to achieve authentication without any intrusion to the user. We believe activity sensor based user identification is less intrusive because it does not require much user interaction with the device thus to make authentication easier.

In recent years, activity sensor-based biometrics has been a hot research subject. The widespread availability of activity sensors such as accelerometers and gyroscopes has resulted in many new designs and technologies aimed at constructing user identities based on sensor data. For identities, previous prototypes have used various movements such as walking, running, jumping, and arm gestures [4, 55, 56]. However, due to the following issues, we have yet to see large-scale implementations of these technologies. For starters, we've seen sensors deployed on various body parts, such as the waist [1, 2], leg [3], sternum [57], wrist [46], and multiple body locations at once [22]. Many of them are unusable in real-world circumstances. Given the increasing popularity of smartwatches (e.g., Apple Watch) and fitness bands (e.g., Fitbit), we assume it is more realistic to create identification using activity data obtained by activity sensors mounted on these devices. We won't need to add any additional sensors to the human body as a result of this process. We will need to reduce the number of sensors in order to make the design cost-effective. Second, the large size of the feature set increases the recognition algorithm's complexity. The feature set must be kept as minimal as possible. On the other hand, we don't want to overlook essential characteristics that contribute to identity uniqueness. It's difficult to correctly classify a consumer based on a limited collection of high-quality features. Third, existing activity sensor-based identification algorithms can still be improved in terms of accuracy. Fourth, while many user identification applications need real-time processing, several embedded devices, such as smart lockers and smart wristbands, are resource constrained, with a slow processor and limited memory. As a result, user recognition algorithms must be lightweight in order to operate on a number of smart devices. Finally, in order to have a seamless user interface and meet real-time specifications, the recognition process should be completed in under a minute. As a result, only a limited amount of data can be obtained.

In [50], kayastha et al. proposed optimal feature evaluation and selection method to evaluate features quantitatively and select high quality features. We believe that there is still scope to improve the filtering of features using correlation method. They also proposed a prototype user identification based on the high quality features. However they have used same session data for training and testing. This needs to be tested in different sessions since training and testing occurs at different times in real case scenario. Also they have used a traditional classifier for user identification by using sliding window method. We believe that applying majority voting to the sliding window method results in improving the accuracy.

The ActID architecture, which consists of a feature evaluation and selection process, a collection of high-quality features from multiple viewpoints, a sliding window based identity modeling algorithm, and a majority voting method addresses the above challenges.

CHAPTER 4

DESIGN OF THE ACTID FRAMEWORK

The purpose of this research is to propose a smooth and non-intrusive way of authenticating user based on activity sensor data collected from the user's wrist. For this, we have designed our ActID Framework. Figure 4.1 illustrates the ActID structure. The framework is divided into two sections: identity modeling and identification.

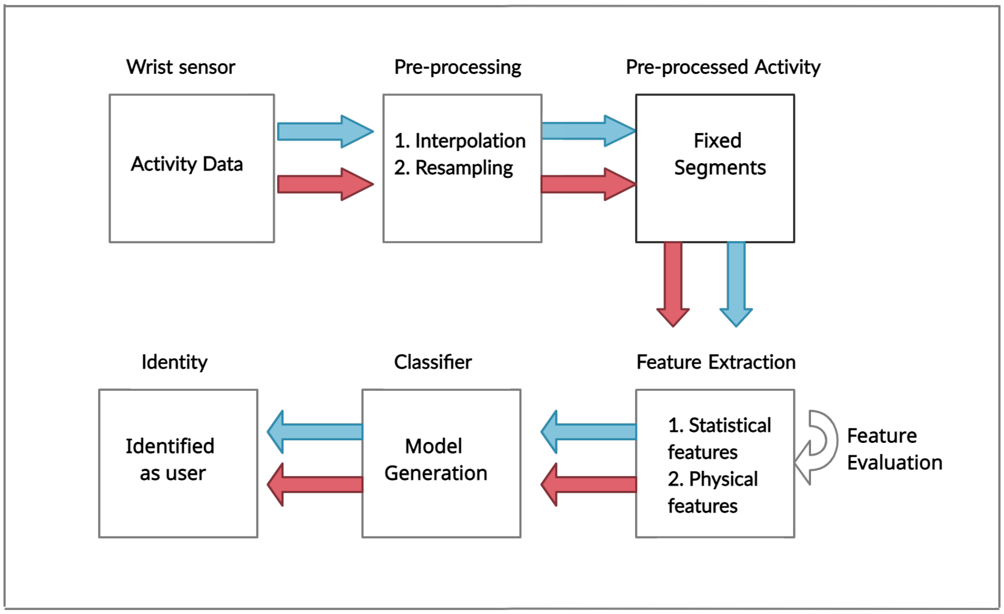


Figure 4.1: *The ActID Framework.*

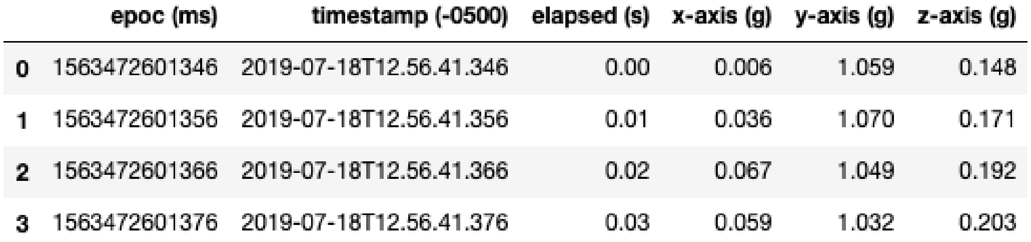
The identity modeling phase is depicted by blue arrows in Figure 4.1, while the recognition phase is depicted by red arrows. When the user walks around in the first step, the changes in motion are captured by an activity monitor, which consists of an accelerometer and a gyroscope and is worn on the user's wrist. The sensing data is then transmitted via Bluetooth to a smart computer. Following that, the data is filtered, resampled, and interpolated to enhance its accuracy. The generated data is used to extract a collection of features. These characteristics include both statistical characteristics such as mean, standard deviation, and variance, as well as physical characteristics such as the peak value for a hand motion acceleration. To test extracted features and choose high-quality features, feature evaluation algorithms such as Optimal Feature Evaluation and Selection (OFES) and Correlation-based Feature Subset Selection (CFSS) are used. Then, using a sliding window algorithm and voting system, we create a Sliding Window based Voting (SWV) classifier as the identity model. Similar to the first phase, user interaction data is obtained in the second phase. Following that, the qualified classifier receives the collected user data as input, and the classifier identifies the user. The specifics of the ActID structure are then presented.

4.1 Data Acquisition

We used the same activity data from 30 users in two sessions, collected by Kayastha et al. [50]. All of these 30 users are student volunteers with nearly similar ages. In each session, users walk as they usually walk on a plain surface for 60 seconds. They used MetaWear C board equipped with two sensors including an accelerometer and a gyroscope, which is placed on the wrist of the user. Sensors capture readings of an accelerometer and gyroscope along x, y, and z-axes. The two sensors captured the hand movement of users as they walk and consist of an accelerometer and gyroscope readings along x, y, and z-axes. Therefore, each data point was a 6-tuple, (Ax, Ay, Az, Gx, Gy, Gz), where A_i and G_i specify an accelerometer and gyroscope on the i axis, respectively. In each of the two sessions, they collected 60 seconds of data sampled at a frequency of 100 Hz. Figure 4.2 represents the sample consisting of 0.03 second of raw data. The two sessions of data are collected in an inter-session environment which is a realistic scenario since enrollment and verification happens at different times.

As shown in phase one, the data obtained in the first session is used to create the classifier. In phase two, the data from the second session will be used to test the classifier. The classifier's output is interpreted as the user's identity.

The sample of an accelerometer data of a user in the X, Y, and Z dimensions for two sessions (S1 and S2) is shown in the Figure 4.3. The blue color lines in the figure reflect data from session 1, while the orange color lines represent data from session 2. Several studies [3, 58] have used a vector summation approach to combine signals from all three dimensions. These methods have the benefit of minimizing computation time by reducing the number of dimensions. However, if the signal amplitude in one dimension is much greater than in others, the signal amplitude in the other dimensions is ignored. We use data in all three dimensions separately for feature computation and comparison in our analysis since it helps in identifying high-quality features.



	epoc (ms)	timestamp (-0500)	elapsed (s)	x-axis (g)	y-axis (g)	z-axis (g)
0	1563472601346	2019-07-18T12.56.41.346	0.00	0.006	1.059	0.148
1	1563472601356	2019-07-18T12.56.41.356	0.01	0.036	1.070	0.171
2	1563472601366	2019-07-18T12.56.41.366	0.02	0.067	1.049	0.192
3	1563472601376	2019-07-18T12.56.41.376	0.03	0.059	1.032	0.203

Figure 4.2: *Samples of Raw Data.*

4.2 Data cleaning and transformation

In [50], Khayastha et al. pre-processed the activity data by interpolation and resampling, and smoothing filter. Resampling is the process of filling the missing data point with the nearest possible value using the linear interpolation method. Khayastha et

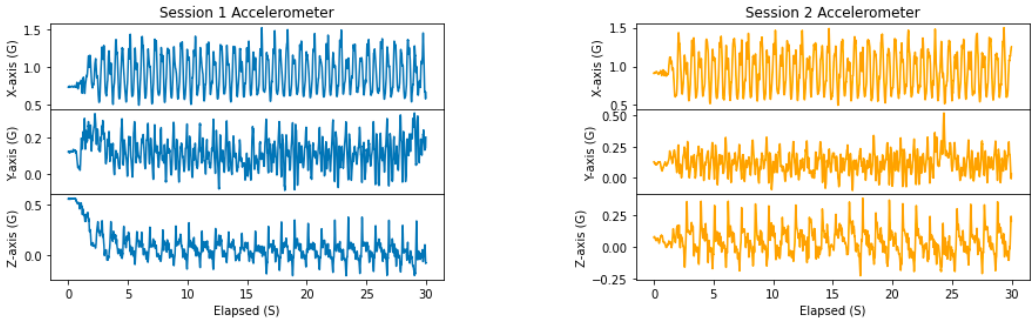


Figure 4.3: *Samples of an accelerometer readings of the same user for two sessions.*

al. have used only data of 14 users for their experiments, whereas we use activity data of 30 users. Since the first few and last few data points may contain more noise, they eliminated the first and the last 2000 data points in the dataset and selected 2000 data points.

We have used the same raw activity data and used linear interpolation and resampling to fill the missing data points of activity data. We have avoided applying the smoothing filter to the activity data since it resulted in the decrease of identification accuracy. We believe that interpolation and resampling is a simple but efficient method for pre-processing. Finally, we have eliminated the first and last few data points as mentioned above.

4.3 Feature Evaluation and Selection

One of the most important steps in developing any biometrics-based identification algorithm is to identify unique features of the biometric dataset [10]. On one hand, the efficiency of the identification algorithm is influenced by the size of the feature set. Typically, to identify a user, a single feature may not be sufficient. Most of the previous studies use a vector of features in their algorithms [22, 14] which increases both the size and dimension of the dataset and results in the increase of complexity of the identification algorithm. On the other hand, the accuracy of the results of identification is primarily influenced by the quality of selected features. We would need high-quality features that can differentiate any two users distinctly which increases the accuracy of the user identification significantly. Distinguishing a particular user from other users is not significant if they are compared using a weak feature. Thus, by excluding weak features based on the results of the feature evaluation, we seek to find a minimum set of high-quality features.

We select a minimum set of high-quality features for gait identification based on the results of our feature evaluation by applying Optimal Feature Evaluation and Selection (OFES) [49, 50, 51] and Correlation-based Feature Subset Selection (CFSS) [52] algorithms. First, we extract biometric features which consist of statistical attributes such as mean, median, and variance, as well as physical attributes like peak value for an acceleration of hand, from the pre-processed raw dataset. OFES provides two of the measures, Farness Value and Farness Ratio to evaluate features [49, 50, 51]. Based on these values, we rank the features according to the ranking method of OFES and identify the high-quality features subset. Second, we reduce the number of sensors used to collect data from users during the identification process in order to make a cost-effective design. To do so, we select only the high-quality features from a sensor that contributes to 70% of high-quality features or more. For example, let's say that among the top 10 selected features, the first 7 features are from the accelerometer, and the last 3 features are from the gyroscope sensor. Since the accelerometer contributes to the majority (i.e., 70%) of the high-quality features, we replace the last 3 features from gyroscope with accelerometer features whose ranks are closest to those 3 gyroscope features. This way we select the 10 high-quality features from the accelerometer sensor only. Third, we apply CFSS to select the set of high-quality features that are correlated to the class label, but independent of each other. To do so, for each feature, we check correlation with every other feature with respect to the class label and identify a set of features that are independent of each other but correlated with the class label.

4.4 Sliding Window Vote (SWV) Classifier

Satisfying the real-time requirements of the user is important in identification. We need to collect only a small set of data from the user so that identification completes in a very short period and provides a smooth user experience. Achieving high accuracy with less amount of data is a challenge in classification. Overcoming this challenge, we design Sliding Window Vote (SWV) Classifier on top of a traditional classifier. To make decisions on a small set of data, SWV utilizes sliding windows which not only helps normalize the data but also helps in reusing data in multiple windows. It also adopts a voting method which helps to improve the accuracy of identification.

The sliding window method solves three issues. First, while comparing two users, it is important to align their activity cycles. Second, a small amount of data will not be sufficient enough to classify a user. With a sliding window, we can generate more windows by overlapping and reusing a set of data. Third, overlapping data between subsequent windows improves the accuracy of the classification.

The design of SWV classifier (also referred to be SWV for the rest of the paper) is depicted in Figure 4.4. It consists of three major components: a set of windows represented using $W_1, W_2, W_3...., W_n$, a traditional classifier, and an aggregator.

The windows are used to hold the data segmented from the sensor data stream using the sliding window approach, the main idea of which is presented in Figure 4.5. As shown in the figure, d_i represents the data points at position i. The sliding window takes the first window of WS data points beginning at position d_1 and ending at

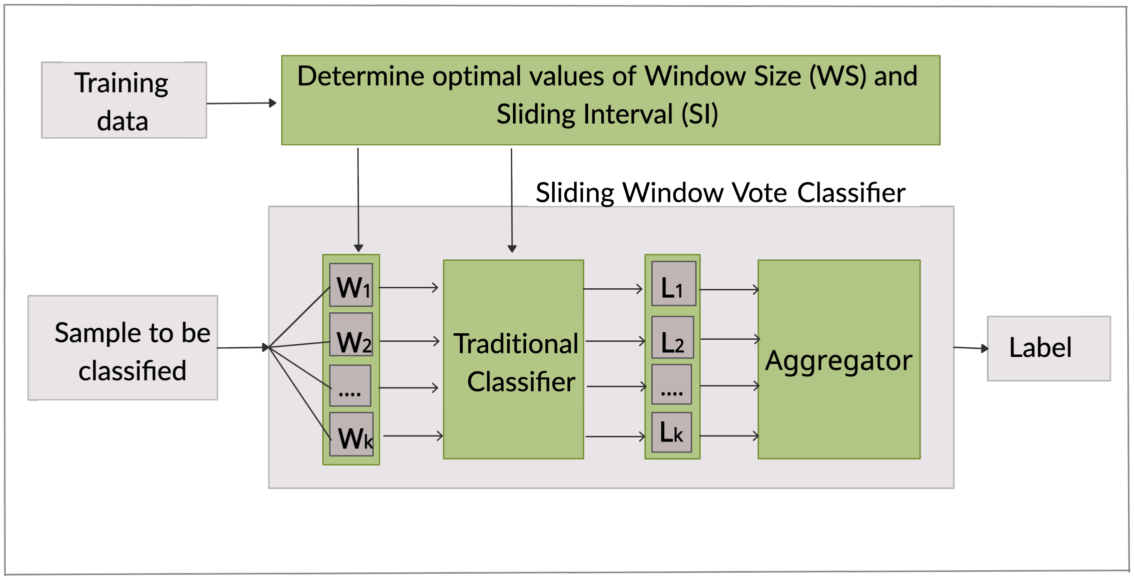

Training data
Determine optimal values of Window Size (WS) and Sliding Interval (SI)
Sliding Window Vote Classifier
Sample to be classified
W1
W2
....
Wk
Traditional Classifier
L1
L2
....
Lk
Aggregator
Label

Figure 4.4: Design of the Sliding Window Vote Classifier (SWV).

position d_{WS}, and places it in W_1 in Figure 4.4. Then, it slides right by SI positions and takes the second window of data starting at position d_{1+SI} and ending at position d_{WS+SI}. This set of data will be placed into W_2. This process will be continued until W_k is filled, which is the last window of the data sequence.

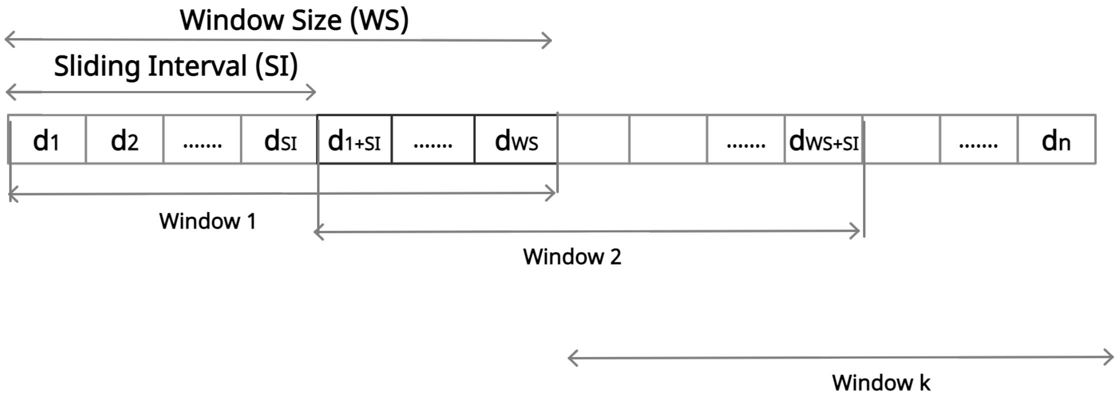


Figure 4.5: Sliding Window Based Feature Extraction.

In the sliding window approach, two parameters, $Window\ Size$ that is defined as the fixed amount of time for how many data points contained in a window, and *Sliding* $Interval$ that is defined as a fixed amount of time for how many data points the window will shift, have a big impact on the performance of the classifier. Therefore, the values of these two parameters should be carefully determined, which is achieved by the process of determining optimal values of window size and sliding interval represented in Figure 4.4. Once the optimal values are determined, they need to be kept the same for the rest of the process.

The traditional classifier can be any existing lightweight classifier such as Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes Classifier. Our SWV classifier is built and optimized based on these classifiers. The choice of classifier also impacts the performance of the SWV classifier.

The last component of SWV is the aggregator. Data in each window will be used as the input to the traditional classifier. Accordingly, a class label will be generated for each window of data. The aggregator generates the final class label by aggregating the class label of each window. Majority voting is used in the process of aggregating. In other words, the aggregator counts the votes for each label and selects the label with the highest number of votes.

SWV is trained using the training dataset, which is produced from session one data collected in Section 4.1. First, we find the optimal values for the parameters such as window size and sliding interval, the process of which is discussed below. Then, we generate windows of data with the optimal values of window size and sliding interval from session one data of activity dataset as discussed before. After that, we extract selected features from each window of data to generate a feature dataset. This feature dataset is used as a training dataset and sent to the traditional classifier component as input to train SWV.

To determine optimal values of Window Size and Sliding Interval, first, we initialize window size with any small value and a fixed size for Sliding Interval e.g., 0.5 seconds, corresponding to 50 data points. As before, we generate a training dataset from session one data of the activity dataset. Then, we train any traditional classifier with a training dataset. Similar to training data, we generate test data from session two data of the activity dataset. Next, we test the classifier using test data to measure accuracy. From our results, we observe that the accuracy of the classifier increases with the increase of window size until a certain point and then decreases. Hence, we increase the window size, extract the feature dataset, and generate the classifier again. We repeat this process until the accuracy of the classifier starts decreasing. Finally, we select the window size which results in the highest accuracy of the classifier. Similarly, we determine the optimal sliding interval that gives high accuracy by fixing the optimal value for window size and increasing the values of the sliding interval starting from 0.01 second (sliding only one data point), 0.25 second, and so on. In the process of determining sliding interval, we observe that the accuracy of the classifier decreases with the increase of sliding interval. However, with a sliding interval of 0.01 second, data generation time is very long and accuracy is only a little higher than in case of 0.25 second. Hence, we select 0.25 second as the optimum sliding interval by trading off the accuracy with the efficiency.

After SWV is trained, it is used to identify the users. In this process, a few seconds of user activity data is the input to the SWV, and the identity of the user will be the output of the classifier.

4.4.1 Evaluation Metrics

We use four metrics commonly used in user identification studies to assess the effectiveness of our proposed method [38, 39].

- False Accept Rate (FAR) is the percentage of identification instances in which unauthorised persons are incorrectly accepted
- False Reject Rate (FRR) is the percentage of identification instances in which authorised persons are incorrectly rejected
- Equal Error Rate (EER) is the intersection of FAR (False Acceptance Rate) and FRR (False Rejection Rate). A device with a lower EER is considered more precise.
- Accuracy (also known as True Positive Rate, or TPR) is the percentage of all identification attempts that correctly identify users.

4.4.2 Classification Algorithm

We use the users' session 1 dataset as our training dataset and the users' session 2 dataset as our test dataset to construct a classifier. Our labels are based on the number of users we have, with User IDs ranging from 1 to 30. Training dataset contains the actual labels for given data points. On the other hand, we delete the labels from the test dataset to see how well a classifier can classify the consumer. To train our classifier model, we use the training dataset. The trained model is then used to classify or predict users on the test dataset.

We send the input of activity data to the classifier in segments of fixed size. For example, we select 10 seconds of activity data in our experiment as optimal size of activity data for the segment for training or testing. In training, we send the activity data of all the users with User ID from 1 to 30 in segments to the SWV classifier. In testing, we send the activity data of any user with User ID from 1 to 30 to the SWV classifier. SWV Classifier outputs with the label of the user. We verify the actual and predicted labels and measure accuracy i.e accuracy would be 100% if matches otherwise zero. Similarly, we test all the 30 users and take the average of accuracy.

We also measure equal error rate as a metric to measure the performance of the classifier. First, we select any user and set the label as positive label (Eg. Label 1), binarize it as label 1 and all other labels as zero. Next, we send segments of test data of both positive label and negative label to the SWV classifier. And, store the percentage of votes of positive label as the predicted score for each data segment. Finally, we send the array of test labels (actual labels of each data segment) and array of predicted scores of positive label to the *roc curve* method [59] from scikit-learn python library [60] to evaluate false positive rate and true positive rate. By sending these measures as parameters to *brentq* method [59], we calculate equal error rate. We repeat this process for all the labels (i.e., binarize different label for each time) and take the average of equal error rate.

CHAPTER 5

PERFORMANCE EVALUATION

In this section, first, we present the performance evaluation of OFES when it is applied to the activity sensor based user identification application. Second, we present the performance evaluation of ActID.

In the rest of this section, we first describe the characteristics of the dataset. Then, we illustrate the performance comparisons between OFES and others. Then, we illustrate the performance comparisons between ActID and others. Finally, we discuss about the effectiveness of OFES and ActID, and applications.

5.1 Description of Dataset

As we discussed in Section 4.2, we select 20 seconds of activity data, i.e., 2000 samples from the processed dataset of 60 seconds of data, i.e., 6000 samples. The training and testing dataset is the feature dataset calculated from the 20 seconds of activity data following the window generation procedure as discussed in Section 4.4. As detailed in Section 5.3.1, we find that 10 high-quality features represent a great trade-off between the classification accuracy and complexity.

Various window sizes and sliding intervals are used in the process of determining their optimal values which are discussed in Section 5.3.1. We determine the optimum window sizes of 6, 8, and 10 seconds for 10, 15, and 20 seconds of activity data respectively, and 0.25 second as an optimum sliding interval in all three cases. After that, the optimal values of window size and sliding interval are used in the rest of the experiment for the specific activity data size. Later, these optimal values of window size and sliding interval are used in the process of generating feature dataset. For each specific feature, we will have two sessions of feature dataset. Session one feature dataset was used as train dataset whereas session two feature dataset was used as test dataset.

5.2 Performance comparison of our feature selection method with other algorithms

Some common feature selection algorithms include ReliefF, Principal Component Analysis (PCA), Correlation Based Feature Selection (CFSS), Information Gain Feature Ranking (IGFR), and Random Projections. Of all the current literature, [46, 38, 39] did a fantastic job of providing detailed work on feature evaluation and selection, which is very relevant to our research. We compare our proposed feature evaluation and selection algorithm with these two best previous efforts by Kumar et al. [46] who used Information Gain Feature Ranking and Damaševičius et al. [39] who used Random Projections with Matlab feature ranking for feature selection. In addition, we also compare OFES with a feature selection approach that randomly selects a set of features from the feature set.

We select top ten ranked features in [46, 39] to compare with top 10 selected features of OFES. In addition, we also compare the performance of classifiers using features selected by above approaches and random selected features. For OFES, feature sets with both 8 feature and 10 feature are used. For the rest of the section, we use OFES(10) to refer OFES selecting 10 features and OFES(8) to refer OFES selecting 8 features.

In the rest of this section, we first compare the differences among top 10 selected featured using OFES, IGFR, and Random Projections with Matlab feature ranking. Then, for each compared feature selection approach, we train classifiers based on four classification algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest Classifier using the set of features selected by that approach. For consistency and fairness, we use the same activity data set for all the methods in order to build classifiers. The performance of these classifiers reflects the performance of these feature evaluation and selection approaches.

5.2.1 Feature evaluation results

Based on our feature evaluation method, we filter the features that have $FV \geq 0.15$ and $FR \geq 15$ as high-quality features from the total of 96 extracted features because the bigger FV and FR implies significant difference between two classes.

From the filtered features based on specific values of FV and FR, we first rank them from the highest to the lowest based on FV and FR separately. Then, we rank them again considering both FV and FR, and select the top 10 and top 8 features out of 96 features.

5.2.2 Comparison of selected feature sets

The authors in [46] have extracted a total of 76 features (32 features from the accelerometer readings and 44 features from the gyroscope readings). They used information gain based feature ranking to rank the features. They ranked the features of accelerometer and gyroscope separately [46] whereas OFES ranks all the features at the same time.

The authors in [39] have extracted 99 features from the collected data based on the extensive analysis of the literature and features used by other authors. They used Random Projections for dimension reduction, and Matlab feature ranking to rank the features.

Rank	**OFES**	**IGFR [46]**	**Random Projections +** **Matlab feature ranking [39]**
1.	Mean ACC X	Median ACC Y	Variance GYRO Z
2.	Median ACC X	Energy ACC X	Variance ACC M
3.	Mean ACC Y	Energy ACC M	First eigenvalue of moving covariance of difference between ACC and GYRO $E_{ag} = eig_1\left(\text{cov}\left(a_x - g_x, a_y - g_y, a_z - g_z\right)\right)$
4.	Median ACC Y	Median ACC Z	Energy GYRO Z
5.	Energy ACC X	Energy GYRO Z	Moving energy of difference between ACC Z and GYRO Z $ME_{ag} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - y_i\right)^2$, here $x = a_z, y = g_z$
6.	Median ACC Z	Median GYRO M	Variance ACC X
7.	Mean ACC Z	Mean Rotation Rate GYRO M	First eigenvalue of moving covariance between ACC $E_a = eig_1\left(\text{cov}\left(a_x(1:N), a_y(1:N), a_z(1:N)\right)\right)$
8.	Energy ACC Y	Energy GYRO M	First eigenvalue of moving covariance between GYRO $E_g = eig_1\left(\text{cov}\left(g_x(1:N), g_y(1:N), g_z(1:N)\right)\right)$
9.	Skew ACC Y	Mean Rotation Rate GYRO Z	Moving energy of orientation vector of ACC $MEA = \frac{1}{N}\sum_{i=1}^{N}\varphi_i^2$, here $\varphi = \frac{\arccos(a_x \cdot a_y)}{\lvert a_x\rvert \cdot \lvert a_y\rvert}$
10.	RMS ACC X	Median ACC X	GYRO M

Table 5.1: *Comparison of top 10 features selected by different feature selection algorithms.*

Table 5.1 shows the comparison of top 10 features selected by different feature selection algorithms. In the table, ACC M and GYRO M are defined as $\sqrt{\left(a_x\right)^2 + \left(a_y\right)^2 + \left(a_z\right)^2}$ and $\sqrt{\left(g_x\right)^2 + \left(g_y\right)^2 + \left(g_z\right)^2}$ respectively. We observe that there are no common features selected by all three algorithms. However there are four common features between OFES and IGFR, including Median ACC X, Median ACC Y, Median ACC Z, and Energy ACC X , and none between OFES and Random Projections with Matlab Ranking. We also observe that OFES selected only accelerometer based features. Hence we can reduce the number of sensors by using only accelerometer sensor and improve cost efficiency of user classification based on activity sensor data.

5.2.3 Performance comparison in scalability to the number of class labels

In multi-class classification, the accuracy of classifier usually decreases with the increase in the number of class labels [61]. Classifiers constructed based on high-qualify features should be scalable with the class labels. That is to say, the classifiers should maintain high accuracy with the increase of the number of class labels.

In this experiment, we compare the performance of five feature selection approaches in 14, 20, 25, 30-class classification. We select four different classifiers including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest Classifier and select the best classifier with high accuracy. The results are depicted in Figure 5.1, where x-axis specifies the number of class labels, and y-axis indicates the accuracy of the best classifier constructed using the features selected by the corresponding feature selection algorithm. Each colored line represents the accuracy of best classifier constructed based on the selected feature set for a specific number of class label.

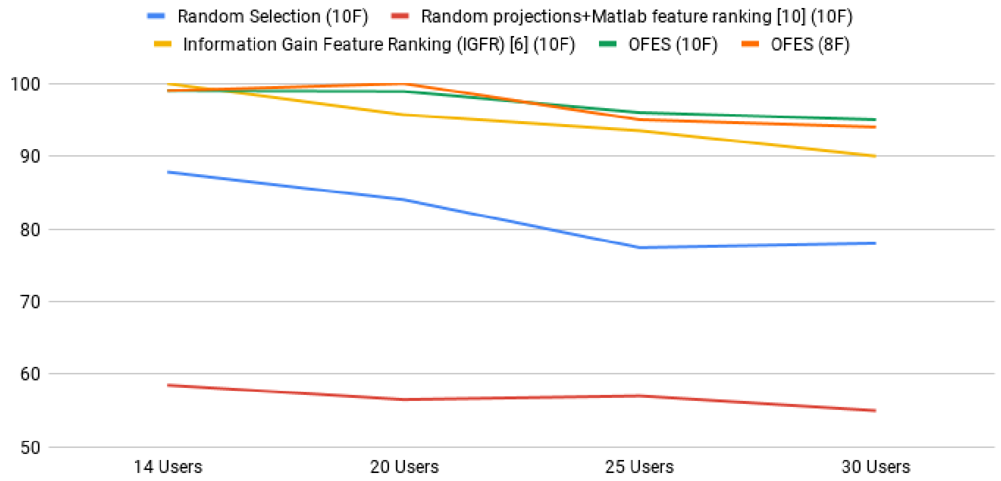


Figure 5.1: *Performance comparison in scalability to the number of class labels.*

From the figure, we observe that accuracy of classification tends to decrease with increase in the number of class labels (users) for all algorithms. Among all approaches, IGFR features results in the most significant accuracy drop, which is 10%, dropping from 100% for 14-user classification to 90% for 30-user classification. Following that is random selection features with a drop of 9.8%. Random Projections with Matlab feature ranking has similar scalability performance to OFES(10) and OFES(8). They all drop about 5% with the increase of number of users. However, the classifiers constructed using OFES(10) and OFES(8) achieve way better accuracy than that using Random Projections with Matlab feature ranking.

5.2.4 Performance comparison in sensitivity to classification algorithms

Features selected with a feature selection algorithm should be less sensitive across different classifier models because each classifier will have its own advantages and disadvantages. Hence, there is a need for selected high-quality features to perform well with any kind of classifier model.

We select four classifier models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest Classifier in order to test the sensitivity of the feature selection algorithms. Figure 5.2 shows the comparison of sensitivity between different classifiers with respect to feature selection algorithms for 30-user classification. In the figure, the x-axis specifies different classification algorithms, and y-axis indicates the accuracy of the classifier constructed based on the corresponding classification algorithm by using the selected feature set. Each line in the figure denotes the accuracy of the classifier constructed by using feature set selected by a specific feature selection approach based on different classification algorithms. We compare the sensitivity based on standard deviation of the accuracy of classifier.

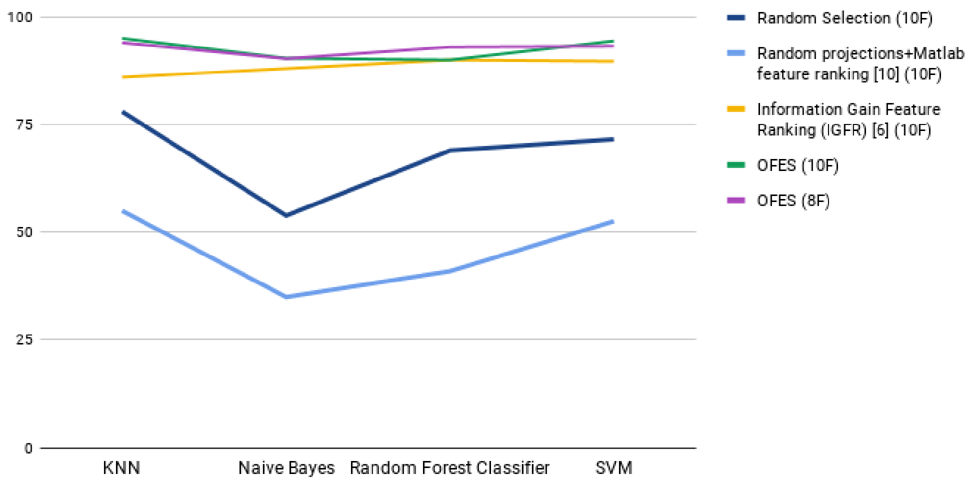


Figure 5.2: *Performance comparison in sensitivity to classification algorithms.*

From the figure, we observe that Random Projections with Matlab feature ranking and random selection features exhibit high standard deviation of 8.22, 8.84 respectively while OFES (10), OFES (8) and IGFR exhibit consistent accuracy results with very small standard deviation of $2.26, 1.39, 1.59$ respectively across different classifier models. The gap between accuracy of the best and the worst classifiers for Random Projections with Matlab feature ranking and random selection features ranges between $14 - 20\%$ while for OFES and IGFR, it ranges between only $3 - 5\%$. This indicates OFES and IGFR are less sensetive to different classification algorithms. Compared with IGFR, OFES has higher accuracy in general. In summary, OFES achieves both consistency and accuracy across different classifiers.

5.2.5 Performance comparison in the impact of the features on accuracy of classifiers

Selecting the high-quality features is necessary to accurately distinguish the classes. We use accuracy of the classifier models as a measure to evaluate the performance of feature selection algorithms. In order to compare the performance of feature selection algorithms, for each feature selection algorithm, we build four different classifiers including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest Classifier for multi-class classification of 30 users with the features selected by the corresponding feature selection algorithm.

5.2.6 Accuracy in 30-user classification

In this section, we compare the overall performance of feature selection algorithms for all the classifier models. Figure 5.3 shows the performance comparison in the impact of the features selected by different feature selection algorithms on accuracy of classifiers in classifying 30 users. In the figure, the x-axis specifies different feature selection algorithms, and y-axis indicates the accuracy of the classifier constructed by using the selected feature set based on the corresponding feature selection algorithm. Each line in the figure denotes the accuracy of the specific classifier constructed by using feature set selected by different feature selection algorithms.

From the figure we observe that Random projections with Matlab ranking results in the least accurate results with an average accuracy of 45.9% among all classifier models. IGFR with an average accuracy of 88.42% exhibits better results than random selection features and Random Projections with Matlab ranking, but it is not as good as OFES. OFES(10) achieves better results when compared to others for all the

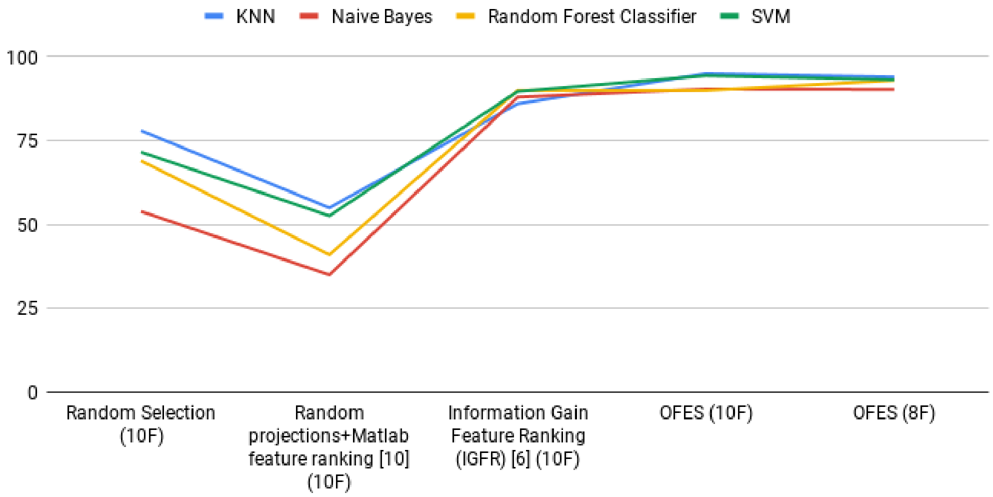


Figure 5.3: *Performance comparison in the impact of the features on classification accuracy.*

selected classifier models with an average accuracy of 92.45%. Even OFES(8) exhibits higher average accuracy of 92.62% when compared to others. We also observe that KNN is the best classification method for the majority of feature selection algorithms.

5.2.7 Best accuracy in 30-user classification

In this section, we compare the classifiers that achieve the highest accuracy for the corresponding feature selection algorithm. Table 5.2 shows the comparison of OFES with other approaches for 30-user classification in terms of accuracy and best classification method. With KNN as the best classification method, random selection features and Random Projections with Matlab ranking achieves accuracy of 78%, 55% respectively, while IGFR achieves accuracy of 90% with Random Forest Classifier as the best classification method. OFES(8) and OFES(10) achieves higher accuracy of 94%, 95% respectively with KNN as the best classification method when compared to others.

From all above analysis, we conclude that OFES outperforms other feature evaluation and selection algorithms as it is less sensitive to different classification algorithms and more scalable to the number of class labels. It also achieves higher accuracy for most classification algorithms by identifying a smaller set of high-quality features.

Feature Selection Algorithm	Best Classification Method	Accuracy
Random selection	KNN	78%
Random Projections + Matlab feature ranking [39]	KNN	55%
Information Gain Feature Ranking (IGFR) [46]	Random Forest Classifier	90%
OFES(8)	**KNN**	**94%**
OFES(10)	**KNN**	**95%**

Table 5.2: *Performance comparison in accuracy of the best classification method.*

5.3 Performance comparisons between ActID and others

In this Section, we first determine the parameters of the SWV classifier such as optimal window size and sliding interval, choice of the best classifier, selection of feature sets, and activity data size. Next, we conduct a performance evaluation of SWV based on the optimal parameters. The evaluation includes performance in the reduced data set, performance of the scalability, and performance in the accuracy. Finally, we perform a comparison between ActID and other frameworks.

5.3.1 Finding optimal values of SWV parameters

Optimal Feature Set In this experiment, we considered 96 features that are used in [49, 50, 51]. We extract these features and apply OFES and CFSS algorithms as we discussed in Section 4. To identify the high-quality features, first we ranked them from highest to lowest based on Farness Value and Farness Ratio [49, 50, 51]. Then, we identify the top 10 features each from the Farness Value and Farness Ratio list. In both of these lists, 8 out of 10 features are in common. Next, we select the top 10 features from both Farness Value and Farness Ratio list considering the ranks of both Farness Value and Farness Ratio. Table 5.3 represents the top 10 features selected. In the table, all the top 10 features belong to accelerometer readings. Hence, we use only one sensor, i.e., an accelerometer to collect activity data during user identification.

Rank	**OFES**	**Definition**
1.	Mean ACC X	Mean of acceleration data along the x-axis. The mean is the most common measure of central tendency. It is simply the sum of the numbers divided by the number of numbers.
2.	Median ACC X	Median of acceleration data along the x-axis. The median is also a frequently used measure of central tendency. The median is the midpoint of a distribution.
3.	Mean ACC Y	Mean of acceleration data along the y-axis.
4.	Median ACC Y	Median of acceleration data along the y-axis.
5.	Energy ACC X	Energy of acceleration data along the x-axis. The total energy of a signal x is defined as the sum of squared moduli.
6.	Median ACC Z	Median of acceleration data along the z-axis.
7.	Mean ACC Z	Mean of acceleration data along the z-axis.
8.	Energy ACC Y	Energy of acceleration data along the y-axis.
9.	Skewness ACC Y	Skewness of acceleration data along the y-axis. Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point. We compute the Skewness by using the scipy.stats.skew library in Python.
10.	RMS ACC X	RMS of acceleration data along the x-axis. The root mean square, also known as the quadratic mean, is a statistical measure of the magnitude of a varying quantity, or set of numbers. Its name comes from its definition as the square root of the mean of the squares of the values.

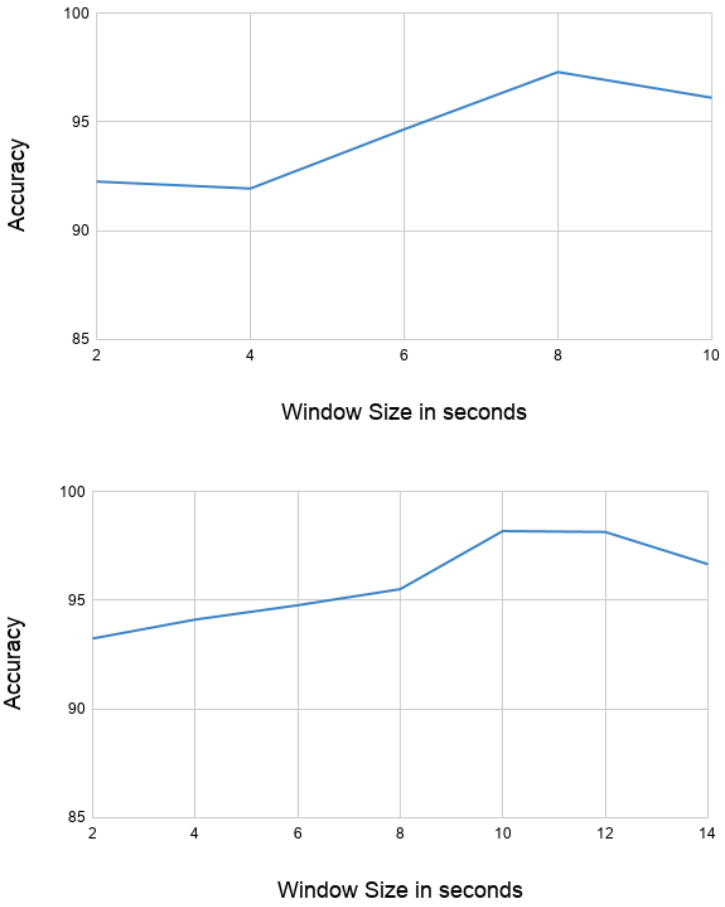
Table 5.3: *Top 10 features selected.*

Optimal Window Size This experiment was conducted to find the optimum window size required to uniquely identify a person. We select SVM classifier as our default standard classifier which is discussed in the following sections. To find the optimal window size, we use various values of window sizes starting from 2 seconds and a fixed size of sliding interval. For example, we use 0.5 second of sliding interval for this experiment.

Figure 5.4 demonstrates the accuracy with different window sizes for 15 and 20 seconds of activity data or data segment respectively. In these figures, the x-axis represents the window size in seconds whereas the y-axis represents the accuracy of the SVM classifier. From both figures, we observe that, since the size of the total dataset is fixed in this study, the graph achieves a peak and then starts to fall. We select the window sizes at the peak point which are 8 seconds and 10 seconds as optimum values in the case of 15 and 20 seconds of activity data respectively.

Optimal Sliding interval Similar to window size, an experiment was conducted to find the optimum sliding interval required to uniquely identify a person. Likewise, when analyzing the impact of sliding interval to the classification accuracy, we fix the window sizes to optimal values of 8 seconds and 10 seconds for 15 and 20 seconds of activity data respectively.

Figure 5.5 demonstrates the accuracy of SVM classifier with different sliding intervals for 15 and 20 seconds of activity data. In the figure, the x-axis represents the sliding interval in seconds whereas the y-axis represents the accuracy of the SVM classifier. We use different values for sliding intervals such as 0.01, 0.25, 0.5, 1 second, and so on. As mentioned in Section 4.4, we skip the sliding interval size of 0.01 second. We observe that since the size of the total dataset is fixed in this study, the accuracy of the classifier decreases with the increase in the sliding interval. As per our analysis,



(a)

(b)

Figure 5.4: *Impact of window size on a) 15 seconds of data and b) 20 seconds of data.*

0.25 second of the sliding interval is an optimum size of the sliding interval in both cases of 10 and 20 seconds of activity data which results in the highest accuracy of the SVM classifier.

Optimal number of features The size of feature set impacts the accuracy as well as time complexity of the classifier. Hence, it is necessary to select a minimum number of high-quality features. We conduct an experiment where a set of classifiers are constructed using various classification algorithms and based on a different number of selected features. We select four lightweight classification algorithms which are

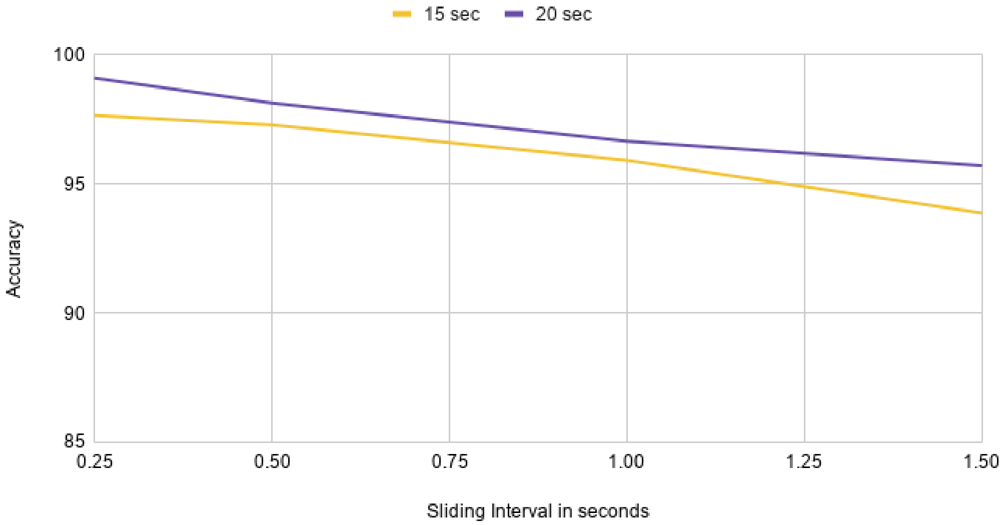


Figure 5.5: *Impact of sliding interval.*

widely used in user identification applications, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Random Forest Classifier. Four different sets of top-ranked features with sizes 8, 10, 12, and 14 are used to construct different classifiers and the accuracy of each classifier is evaluated.

Figure 5.6 shows the results of the above experiment, where the x-axis specifies the size of the feature set, and the y-axis indicates the accuracy of the classifier constructed using a different number of selected features. Each colored line represents the accuracy of a classifier constructed based on a different classification algorithm for a specific number of selected features. From the figure, we observe that classifiers built based upon 4 different sets of features exhibit close performance in terms of accuracy, which is mostly between 90% and 100% except for the Random Forest Classifier which results in around 85%. Classifiers built based on 8, 12, and 14 features have a very close performance for all four classification algorithms, while the classifier on 10 features has slightly higher accuracy. Hence, we select 10 feature set as the minimum feature set that results in high accuracy. Among the four classification algorithms, the

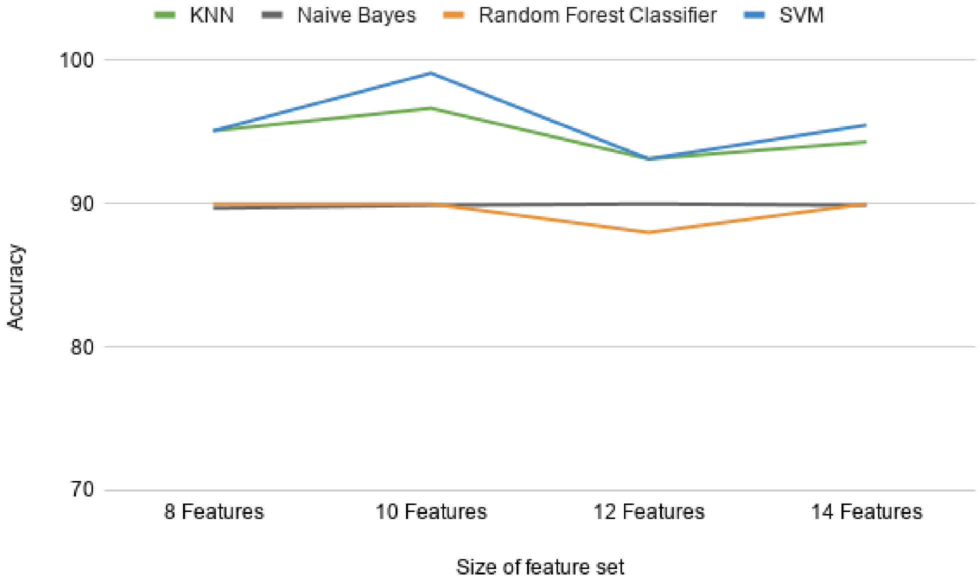


Figure 5.6: *Impact of number of features.*

SVM exhibits the best accuracy while Random Forest Classifier has the least accuracy. Some classification algorithms like Naive Bayes and Random Forest Classifier are less sensitive to the number of features. The above observations confirm our belief that a small number of high-quality features is sufficient to build a highly accurate classifier. It is also necessary to identify a set of high-quality features to reduce the complexity of the classification process.

Optimal Classifier Our SWV classifier is built on top of a traditional classifier. Therefore, the choice of different classifiers may impact the performance of our voting classifier. We believe that deep learning classifiers are too heavy for real-time user identification. Hence, we test four popular lightweight classifiers including KNN, SVM, Naive Bayes, and Random Forest Classifier. We compute the accuracy of the SWV Classifier built on top of these traditional classifiers for comparison.

Classifier Model	KNN	Naive Bayes	RFC	SVM
SWV	96.66%	96.66%	86.66%	100%
Standard Classifier	96.53%	94.76%	85.76%	97.98%

Table 5.4: *Performance comparison between SWV and traditional classifiers in terms of accuracy.*

A comparison between standard classifier and SWV classifier built on top of respective standard classifier can be found in Table 5.4. The results mentioned in the table represents the accuracy of the multi-class classifier that classifies 30 users with 20 seconds of activity data each. SWV classifier with traditional classifier as Random Forest Classifier achieves the least accuracy of 86.66% whereas KNN, SVM, and Naive Bayes results in the accuracy of 96.66%, 100%, and 96.66% respectively. For all the four traditional classifiers, the SWV classifier improves the accuracy. We select the SVM classifier as the best traditional classifier for the SWV classifier since it achieves the highest accuracy compared to others.

5.3.2 Performance evaluation of SWV

From the above experiment, we select SVM classifier as our standard classifier to build SWV classifier, SWV-SVM, in terms of accuracy, scalability, and stability when applied to a small dataset.

Performance comparison between SWV and traditional classifiers Figure 5.7 represents the performance of SWV-SVM with other traditional classifiers. In the figure, the x-axis represents different classifiers whereas the y-axis represents the accuracy of the classifier for 20 seconds of activity data. Random Forest Classifier achieves the least accuracy of 85.76% whereas KNN, SVM, and Naive Bayes results in the accuracy of 96.53%, 97.98%, and 94.76% respectively. SWV-SVM achieves the highest accuracy of 100% when compared to others.

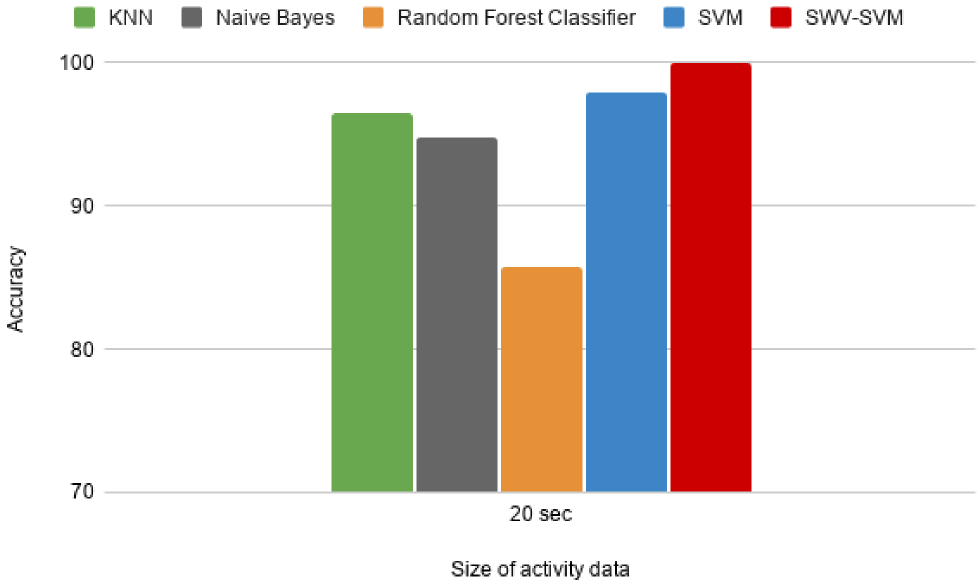


Figure 5.7: *Performance comparison between SWV classifier and other traditional classifiers.*

Scalability of SWV The accuracy of classifiers usually decreases with the increase in the number of class labels in a multi-class classification [61]. Classifiers constructed based on high-quality features should be scalable with the number of class labels. In other words, classifiers should maintain high accuracy with the increase in the number of class labels.

In this experiment, we compare the performance of the SVM classifier (the best-performed classifier among the four evaluated traditional classifiers) and SWV-SVM in 14, 20, 25, and 30-class classification. The results are depicted in Figure 5.8, where the x-axis specifies the number of class labels, and the y-axis indicates the accuracy of the classifier for 20 seconds of activity data. Blue color represents SVM classifier where red color represents SWV-SVM classifier.

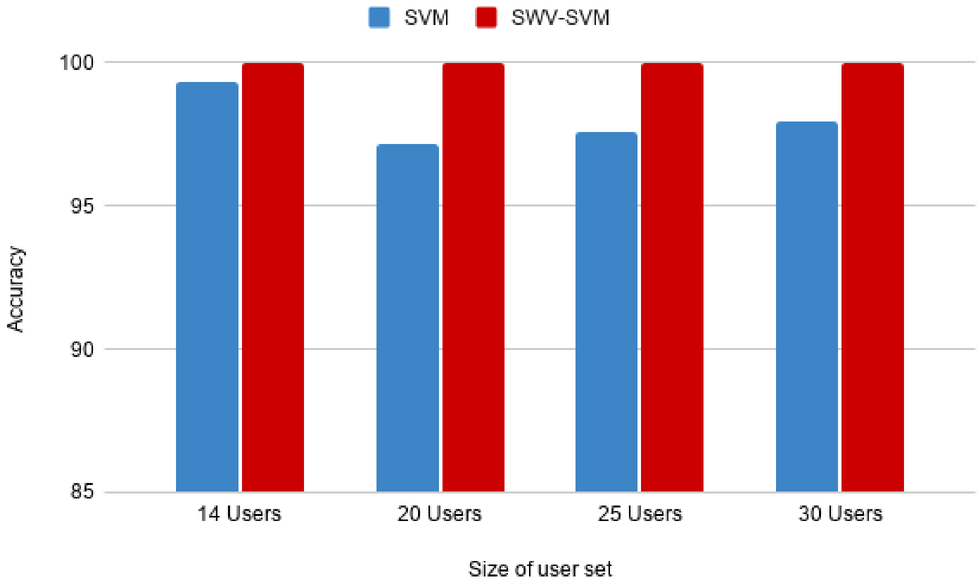


Figure 5.8: *Scalability of SWV.*

From the figure, we observe that, for all datasets with 14, 20, 25, and 30 users, SWV-SVM results in consistent high accuracy while traditional SVM classifier's accuracy decreases with the increase of the number of users. Similar results are seen using other traditional classifiers as well. It shows that SWV not only improves the accuracy of traditional classifiers, but it is also scalable to the size of labels.

Stability of SWV when applied to a small dataset Increasing the size of collected data will both result in a longer data process and longer data collection time, and cause inconvenience to the user. However, smaller activity data size may not capture the entire cycle of walking. Therefore, an optimum size of activity data is required.

Figure 5.9 demonstrates the accuracy of the SVM and SWV-SVM with 10, 15, and 20 seconds of activity data. In the figure, the x-axis represents the activity data size in seconds whereas the y-axis represents the accuracy of classifiers. We observe that SVM trained with 15 seconds and 20 seconds of activity data results in similar performance with an accuracy of 97.7% and 97.96% respectively whereas SVM trained with 10 seconds of activity data results in slightly lesser accuracy of 95.74%. SWV classifier results in 100% accuracy in all three cases. In general, the accuracy of traditional classifiers decreases when the dataset size gets smaller, e.g., 10 seconds. SWV exhibits a better performance than the traditional classifier. As shown in the figure, it not only always has a better performance than traditional SVM but also remains 100% accurate even when the dataset size is reduced to 10 seconds. This helps to achieve real-time authentication.

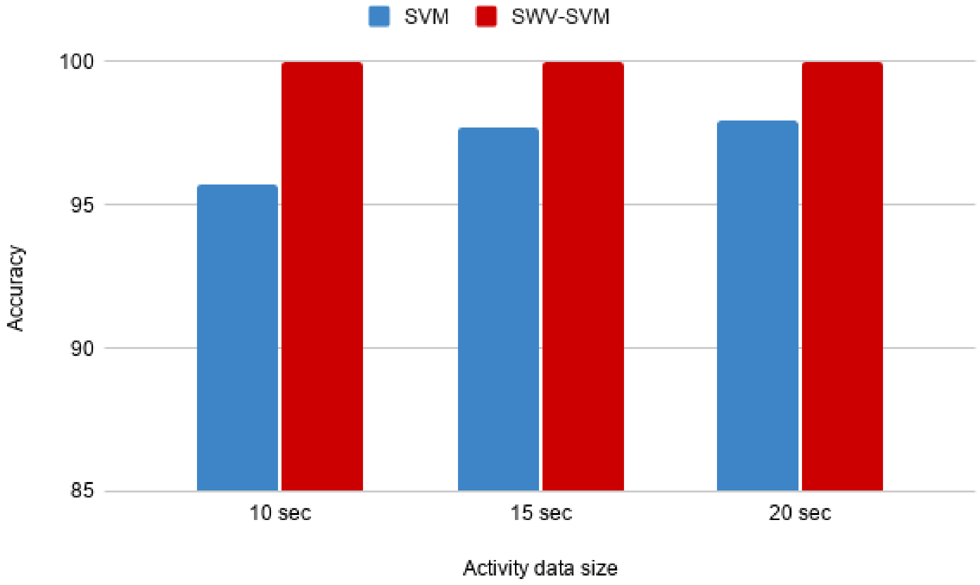


Figure 5.9: *Impact of activity data size.*

5.3.3 ActID with other similar user identification approaches

In this section, we compare ActID with other similar user identification approaches. Table 5.5 shows the comparison of ActID with others in terms of a number of features, best classification method, user set size, activity data size, and accuracy. In the table, two measures including EER, and accuracy are used to show the results. EER is defined as the equal error rate which indicates that the proportion of false acceptances is equal to the proportion of false rejections. The lower the EER, the higher the accuracy of the identification.

In the table, [4] uses the highest number of 52 features with EER 8.24% whereas [3] uses only one feature with EER 5% and 9%. We observe that both the highest and least number of features result in a significant decrease in accuracy. ActID uses an optimum number of 10 features which results in the highest accuracy of 100%. Along with accuracy, we have also measured the equal error rate when applied to the 30 user dataset as mentioned in Section 4.4.2 and ActID results in an EER equal to zero. [28] uses 5 minutes of activity data and results in a lesser accuracy of 84%. [22] results in low EER of 1.23% to 4.07% but collects 4.5 minutes of activity data. [32] collects 20 minutes of activity data and results in lesser accuracy of 86.7%. ActID only uses 10 seconds of activity data but results in the highest accuracy of 100%. [4] and [46] select K-NN as the best classification method, while ActID selects SWV-SVM. [28] uses highest user set size of 59, [32] uses lowest user set size of 7, whereas ActID uses a user set size of 30. [39] and [46] results in accuracy of around 95% whereas ActID results in the highest accuracy of 100%.

In summary, ActID uses an optimum number of features, i.e., 10 features, least amount of activity data, i.e., 10 seconds, yet results in the highest accuracy of 100% when compared to others.

Paper	**Features**	**Best Classification Method**	**Size of User Set**	**Activity data size**	**Results**
[46]	31 features	K-NN	12	2 min	Accuracy: 95%
[39]	10 features	Heuristic (random projections + PDFs + Jaccard distance)	14		Accuracy: 95.52%
[32]	20 features (time and frequency-domain)	C4.5 decision tree classifier	7	20 min	Accuracy: 86.7%
[28]	6 features	Rotation Forest	59	5 min	Accuracy: 84%
[4]	52 features	K-NN	20	1.7 min	EER: 8.24%
[22]	3 types of features	Manhatten method	23	4.5 min	EER: 1.23% to 4.07%
[3]	1 feature	Histogram Similarity and Cycle Length methods	21		EER: 5%, 9%
Our Approach	**10 features (time and frequency-domain)**	**SWV-SVM**	**30**	**10 sec**	**Accuracy: 100%**

Table 5.5: *Comparison of our approach versus other approaches.*

5.3.4 Discussion

Stability of High-Quality Features We believe the results of our feature selection method are consistent across a different number of users. To verify this hypothesis, we compare the top 15 selected features obtained based on 14-user and 30-user datasets. We have the following two observations. First, all the top 15 features are from the accelerometer sensor. This supports our hypothesis that an accelerometer alone may be sufficient for identifying users based on their behavior. Second, 12 out of the top 15 resulted features are in common. This supports our hypothesis that the top-ranked features from our feature selection process are consistent with gait characteristics in different individuals. Table 5.6 lists the 12 common features.

Analysis of the Perfect Accuracy The accuracy results published in this study are based on a 30-user dataset, using our top 10 selected features as well as the optimal values of model parameters, including window size, sliding interval, and data

No.	Features
1.	Energy ACC X
2.	Energy ACC Y
3.	Energy ACC Z
4.	Variance ACC X
5.	Variance ACC Y
6.	Variance ACC Z
7.	Mean ACC X
8.	Mean ACC Y
9.	Median ACC X
10.	Median ACC Y
11.	Root Mean Square ACC X
12.	Root Mean Square ACC Z

Table 5.6: *List of high quality features*

segment size. Many factors may impact the accuracy, including the number of users, the types of users, the choice of parameters, etc., so we believe, for larger size user sets, accuracy may not always be 100% but it could still be very close to 100%, because in our experiments all participants are college students who may have similar activity patterns, which can be considered as a challenging case for identification. In the future, we plan to verify the results of both the feature selection method and the SWV classifier (based on features selected from our feature selection method) with a diversified and large number of user sets.

The two sessions of user data are collected at separate times as we tried to avoid unnecessary similarity introduced in the data collection process. However, we also have to agree that the changes in user's moving patterns over time may have an impact on the identification accuracy. We are currently investigating new approaches that can cope with the pattern changes. This is our future work.

Efficiency Analysis of the Classification Algorithm We believe our classification algorithm is lightweight. First, because of the feature selection algorithm, we can significantly reduce the number of features. This reduces the complexity of the algorithm while maintaining high accuracy. The feature evaluation is done before the classifiers are trained and it only needs to be done once. We can perform feature evaluation on a powerful device such as at the computing edge. Second, the classifier training phase can be separated from the identification phase. The training phase is more computing-intensive than the identification phase. Third, the identification phase is only based on a small amount of data, 10 seconds of activity data. Fourth, we can adjust the sliding intervals to keep the classification phase even more lightweight; however, the impact of accuracy also needs to be considered. All the above designs make the algorithm to be a lightweight algorithm. To verify these arguments, we conducted a preliminary experiment to evaluate the computing cost of the proposed algorithm in terms of execution time as summarized below.

In the experiment, we first evaluated the execution time of identification on an old Macbook Air (Early 2015 model) with a 1.6 GHz Intel Core 5 processor (64-bit dual-core) and 8GB memory size. The identification only took 4 milliseconds. If we consider more computing-intensive tasks, feature extraction and classifier training, the execution time is 2 seconds and 33 milliseconds respectively. Although we were not able to find a direct performance comparison between the processing speed of Macbook Air (early 2015 model) and Apple Watch 6, we found a performance comparison between MacBook Air (early 2015 model) processor and Snapdragon 200, as well as Apple Watch 6 processor and Snapdragon 200. This enables us to have an indirect comparison. The report from Notebookcheck [62] said the processor of Apple Watch 6 is comparable to Snapdragon 200, while the processor of MacBook Air (early 2015 model) is 10 times as fast as the processor of Motorola Moto E i.e., which uses Snapdragon 200 [63]. Therefore, we estimate that the execution time in the smartwatch (Apple Watch 6) would be approximately 20 seconds for feature extraction, 330 milliseconds for classifier training, and 40 milliseconds for identification. When we offload the feature extraction and classifier training to a smartphone like the iPhone 12, which executes 3 times as fast as MacBook Air (early 2015 model) [63], the estimated execution time will be less than 1 second for feature extraction and 11 milliseconds for classifier training. In conclusion, we believe our algorithm is sufficiently lightweight to be executed on mobile devices, even for smartwatches like Apple Watch 6, especially when we offload the heavy computing tasks to an edge device like the iPhone.

Currently, we are developing a continuous authentication protocol based on both smartwatch and smartphone. We will build a prototype to quantitatively evaluate the CPU utilization rate, communication cost, and power consumption.

Applications We believe that activity-based gait recognition can be used in a variety of applications, including multi-factor authentication, where it can be used for continuous authentication, to grant access to a particular room assigned to an individual in a building instead of using key cards to unlock, and so on.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

In this study, we proposed a novel ActID framework that selects only a minimum number of high-quality features, and effectively addresses various real-time challenges of user authentication based on activity sensor data. We introduced a novel Sliding Window Vote Classifier which significantly improved the identification accuracy over traditional classifiers. It demonstrates that even a small amount of activity data and optimal feature dataset is sufficient to uniquely identify a user. This suggests that a balance can be achieved between computation time and accuracy while designing an identification protocol. Furthermore, the SVM classifier is shown to be the consistent and best classifier among the traditional classifiers for user identification based on activity sensor data. Our empirical analysis provided a mechanism to determine the optimal window size and sliding interval, and to reduce the number of sensors used to collect activity data. We demonstrated validity and necessity of finding the parameters such as optimal window size and optimal sliding interval for a specific segment size of activity data in sliding window based feature extraction. Our performance evaluation results show the promising results of activity based user identification.

6.2 Future Work

In the future, we plan to extend the ActId framework to continuous authentication applications and evaluate several factors such as a large number of user sets, diversified user sets, spoofing, etc. We would also introduce a mechanism to learn the biometric changes of the user that occur as the user ages along with a method to detect various activities of the user like walking, running, sitting, etc.

REFERENCES

[1] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, S.-M. Makela, Identifying people from gait pattern with accelerometers, in: A. K. Jain, N. K. Ratha (Eds.), Biometric Technology for Human Identification II, Vol. 5779, International Society for Optics and Photonics, SPIE, 2005, pp. 7 – 14. doi:10.1117/12.603331.

[2] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. . Makela, H. A. Ailisto, Identifying users of portable devices from gait pattern with accelerometers, in: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., Vol. 2, 2005, pp. ii/973–ii/976 Vol. 2. doi:10.1109/ICASSP.2005.1415569.

[3] D. Gafurov, K. Helkala, S. Torkjel, Biometric gait authentication using accelerometer sensor, Journal of Computers 1 (11 2006). doi:10.4304/jcp.1.7.51-59.

[4] C. Nickel, T. Wirtl, C. Busch, Authentication of smartphone users based on the way they walk using k-nn algorithm, 2012, pp. 16–20. doi:10.1109/IIH-MSP.2012.11.

[5] C.-L. Lin, T. Hwang, A password authentication scheme with secure password updating, Computers Security 22 (2003) 68–72. doi:10.1016/S0167-4048(03)00114-7.

[6] R. C. Merkle, A digital signature based on a conventional encryption function, in: C. Pomerance (Ed.), Advances in Cryptology — CRYPTO '87, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988, pp. 369–378.

[7] G. E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Proceedings of the 44th Annual Design Automation Conference, DAC '07, Association for Computing Machinery, New York, NY, USA, 2007, p. 9–14. doi:10.1145/1278480.1278484.

[8] W. Tan, J. Hsu, F. Pinn, Method and system for token-based authentication, US Patent App. 09/792,785 (2001).

[9] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baaaharun, K. Sakurai, Authentication in mobile cloud computing: A survey, Journal of Network and Computer Applications 61 (2016) 59–80. doi:10.1016/j.jnca.2015.10.005.

[10] M. O. Derawi, C. Nickel, P. Bours, C. Busch, Unobtrusive user-authentication on mobile phones using biometric gait recognition, in: 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010, pp. 306–311. doi:10.1109/IIHMSP.2010.83.

[11] K. Sha, M. Kumari, Patient identification based on wrist activity data, in: Proceedings of the 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE '18, 2018, p. 29–30. doi:10.1145/3278576.3278590.

[12] Face id security [online], https://www.apple.com/ca/business-docs/FaceID_Security_Guide.pdf (November 2017).

[13] Windows hello face authentication [online], https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/windows-hello-face-authentication (2017).

[14] D. J. Ohana, L. Phillips, L. Chen, Preventing cell phone intrusion and theft using biometrics, in: 2013 IEEE Security and Privacy Workshops, 2013, pp. 173–180. doi:10.1109/SPW.2013.19.

[15] M. Dey, N. Dey, S. K. Mahata, S. Chakraborty, S. Acharjee, A. Das, Electrocardiogram feature based inter-human biometric authentication system, in: 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014, pp. 300–304. doi:10.1109/ICESC.2014.57.

[16] A. S. Chatra, Cognitive biometrics based on eeg signal, in: 2014 International Conference on Contemporary Computing and Informatics (IC3I), 2014, pp. 374–376. doi:10.1109/IC3I.2014.7019605.

[17] R. P. Wildes, Iris recognition: an emerging biometric technology, Proceedings of the IEEE 85 (9) (1997) 1348–1363. doi:10.1109/5.628669.

[18] H. Zhang, D. Hu, A palm vein recognition system, in: 2010 International Conference on Intelligent Computation Technology and Automation, Vol. 1, 2010, pp. 285–288. doi:10.1109/ICICTA.2010.425.

[19] H. Lee, J. Y. Hwang, D. I. Kim, S. Lee, S.-H. Lee, J. S. Shin, Understanding keystroke dynamics for smartphone users authentication and keystroke dynamics on smartphones built-in motion sensors, Sec. and Commun. Netw. 2018 (2018) 2. doi:10.1155/2018/2567463.

[20] J. Casanova, C. Ávila, G. Bailador, A. Sierra, Authentication in mobile devices through hand gesture recognition, International Journal of Information Security 11 (2012) 65–83.

[21] F. T. Garcia, K. Krombholz, R. Mayer, E. Weippl, Hand dynamics for behavioral user authentication, in: 2016 11th International Conference on Availability, Reliability and Security (ARES), 2016, pp. 389–398. doi:10.1109/ARES.2016.107.

[22] S. K. Al Kork, I. Gowthami, X. Savatier, T. Beyrouthy, J. A. Korbane, S. Roshdi, Biometric database for human gait recognition using wearable sensors and a smartphone, in: 2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART), 2017, pp. 1–4. doi:10.1109/BIOSMART.2017.8095329.

[23] J. Chen, Gait correlation analysis based human identification, TheScientificWorldJournal 2014 (2014) 168275. doi:10.1155/2014/168275.

[24] L. Rong, Z. Jian-zhong, L. Ming, H. Xiang-feng, A wearable acceleration sensor system for gait recognition, 2007 2nd IEEE Conference on Industrial Electronics and Applications (2007) 2654–2659.

[25] H. Sun, T. Yuao, Curve aligning approach for gait authentication based on a wearable accelerometer, Physiological measurement 33 (2012) 1111–20. doi:10.1088/0967-3334/33/6/1111.

[26] J. R. Kwapisz, G. M. Weiss, S. A. Moore, Cell phone-based biometric identification, in: 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2010, pp. 1–7. doi:10.1109/BTAS.2010.5634532.

[27] H. M. Thang, V. Q. Viet, N. Dinh Thuc, D. Choi, Gait identification using accelerometer on mobile phone, in: 2012 International Conference on Control, Automation and Information Sciences (ICCAIS), 2012, pp. 344–348. doi:10.1109/ICCAIS.2012.6466615.

[28] A. H. Johnston, G. M. Weiss, Smartwatch-based biometric gait recognition, in: 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2015, pp. 1–6. doi:10.1109/BTAS.2015.7358794.

[29] X. Su, H. Tong, P. Ji, Activity recognition with smartphone sensors, Tsinghua Science and Technology 19 (3) (2014) 235–249. doi:10.1109/TST.2014.6838194.

[30] A. Primo, V. V. Phoha, R. Kumar, A. Serwadda, Context-aware active authentication using smartphone accelerometer measurements, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 98–105. doi:10.1109/CVPRW.2014.20.

[31] H. Xu, J. Liu, H. Hu, Y. Zhang, Wearable sensor-based human activity recognition method with multi-features extracted from hilbert-huang transform, Sensors 16 (12) (2016) 2048. doi:10.3390/s16122048.

[32] B. Liu, H. Luo, C. W. Chen, A novel authentication scheme based on acceleration data in wban, in: 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2017, pp. 120–126. doi:10.1109/CHASE.2017.70.

[33] D. Sugimori, T. Iwamoto, M. Matsumoto, A study about identification of pedestrian by using 3-axis accelerometer, in: 2011 IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications, Vol. 2, 2011, pp. 134–137. doi:10.1109/RTCSA.2011.64.

[34] D. Guan, W. Yuan, A. Sarkar, Review of sensor-based activity recognition systems, IETE Technical Review 28 (2011) 418. doi:10.4103/0256-4602.85975.

[35] J. Blasco, T. M. Chen, J. Tapiador, P. Peris-Lopez, A survey of wearable biometric recognition systems, ACM Comput. Surv. 49 (3) (Sep. 2016). doi:10.1145/2968215.

[36] M. D. Marsico, A. Mecca, A survey on gait recognition via wearable sensors, ACM Comput. Surv. 52 (4) (Aug. 2019). doi:10.1145/3340293.

[37] C. Yang, D. Liang, C. Chang, A novel driver identification method using wearables, in: 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), 2016, pp. 1–5. doi:10.1109/CCNC.2016.7444722.

[38] J. M. W. Robertas Damaševičius, Mindaugas Vasiljevas, Human activity recognition in aal environments using random projections, Computational and Mathematical Methods in Medicine 2016 (2016) 17. doi:10.1155/2016/4073584.

[39] R. Damaševičius, R. Maskeliūnas, A. Venčkauskas, M. Woźniak, Smartphone user identity verification using gait characteristics, Symmetry 8 (10) (2016) 100. doi:10.3390/sym8100100.

[40] D. López-Fernández, F. J. Madrid-Cuevas, A. Carmona-Poyato, R. Muñoz Salinas, R. Medina-Carnicer, Multi-view gait recognition on curved trajectories, in: Proceedings of the 9th International Conference on Distributed Smart Cameras, ICDSC '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 116–121. doi:10.1145/2789116.2789122.
URL https://doi.org/10.1145/2789116.2789122

[41] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, Z. Yu, Sensor-based activity recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (6) (2012) 790–808. doi:10.1109/TSMCC.2012.2198883.

[42] D. Gafurov, E. Snekkenes, Gait recognition using wearable motion recording sensors, EURASIP Journal on Advances in Signal Processing 2009 (2009) 1–16.

[43] S. Z. Syed Idrus, E. Cherrier, C. Rosenberger, P. Bours, Soft biometrics for keystroke dynamics: Profiling individuals while typing passwords, Computers Security 45 (2014) 147–155. doi:https://doi.org/10.1016/j.cose.2014.05.008.
URL https://www.sciencedirect.com/science/article/pii/S0167404814000893

[44] H. Saevanee, N. Clarke, S. Furnell, V. Biscione, Continuous user authentication using multi-modal biometrics, Comput. Secur. 53 (C) (2015) 234–246. doi:10.1016/j.cose.2015.06.001.
URL https://doi.org/10.1016/j.cose.2015.06.001

[45] H. Crawford, K. Renaud, T. Storer, A framework for continuous, transparent mobile device authentication, Computers Security 39 (2013) 127–136. doi:https://doi.org/10.1016/j.cose.2013.05.005.
URL https://www.sciencedirect.com/science/article/pii/S0167404813000886

[46] R. Kumar, V. V. Phoha, R. Raina, Authenticating users through their arm movement patterns (2016). arXiv:1603.02211.

[47] S. Mondal, P. Bours, Swipe gesture based continuous authentication for mobile devices, in: 2015 International Conference on Biometrics (ICB), 2015, pp. 458–465. doi:10.1109/ICB.2015.7139110.

[48] M. Abuhamad, A. Abusnaina, D. Nyang, D. Mohaisen, Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A

contemporary survey, IEEE Internet of Things Journal 8 (1) (2021) 65–84. doi:10.1109/JIOT.2020.3020076.

[49] N. Kayastha, K. Sha, Poster abstract: A novel and efficient approach to evaluate biometric features for user identification, in: 2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2019, pp. 21–22. doi:10.1109/CHASE48038.2019.00016.

[50] N. Kayastha, Biometrics-based user identification with optimal feature evaluation and selection, Master's thesis, College of Science and Engineering, University of Houston-Clear Lake (December 2019).

[51] V. S. Sai Ram, N. Kayastha, K. Sha, OFES: Optimal feature evaluation and selection for multi-class classification, UHCL Technical Report 2020-01 (2020).

[52] M. A. Hall, Correlation-based feature selection for machine learning, Tech. rep. (1999).

[53] V. S. Sai Ram, N. Kayastha, K. Sha, ActID: An efficient framework for activity sensor based user identification, UHCL Technical Report 2021-01 (2021).

[54] J. R. Vacca, Biometric Technologies and Verification Systems, Butterworth-Heinemann, USA, 2007.

[55] J. P. Gupta, N. Singh, P. Dixit, V. B. Semwal, S. R. Dubey, Human activity recognition using gait pattern, Int. J. Comput. Vis. Image Process. 3 (3) (2013) 31–53. doi:10.4018/ijcvip.2013070103.

[56] A. Abate, M. Nappi, S. Ricciardi, I-am: Implicitly authenticate me person authentication on mobile devices through ear shape and arm gesture, IEEE

Transactions on Systems, Man, and Cybernetics: Systems PP (2017) 1–13. doi:10.1109/TSMC.2017.2698258.

[57] E. Vural, S. Simske, S. Schuckers, Verification of individuals from accelerometer measures of cardiac chest movements, in: 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG), 2013, pp. 1–8.

[58] W. Yang, S. Wang, J. Hu, G. Zheng, J. Chaudhry, E. Adi, C. Valli, Securing mobile healthcare data: A smart card based cancelable finger-vein bio-cryptosystem, IEEE Access 6 (2018) 36939–36947. doi:10.1109/ACCESS.2018.2844182.

[59] C. blog, How to compute equal error rate (eer) on roc curve (Sep 2016). URL https://yangcha.github.io/EER-ROC/

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[61] A. Kumari, U. Thakar, Hellinger distance based oversampling method to solve multi-class imbalance problem, in: 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), 2017, pp. 137–141. doi:10.1109/CSNT.2017.8418525.

[62] K. Hinum, Apple s6 processor - benchmarks and specs (Nov 2020). URL https://www.notebookcheck.net/Apple-S6-Processor-Benchmarks-and-Specs.502601.0.html

[63] Benchmarks, Primate Labs Inc., (Accessed: 9 April 2021). URL https://browser.geekbench.com/