Bachelor's project

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Sindre Toft Nordal

Biometric authentication of smartphone users using accelerometer data

Bachelor's project in computer engineering
Supervisor: Bjarte M. Østvold, Norwegian Computing Center,
Edvard K. Karlsen, Kantega, and Ole Christian Eidheim, IDI
NTNU
May 2019

NTNU
Norwegian University of
Science and Technology

Sindre Toft Nordal

Biometric authentication of smartphone users using accelerometer data

Bachelor's project in computer engineering
Supervisor: Bjarte M. Østvold, Norwegian Computing Center, Edvard K. Karlsen, Kantega, and Ole Christian Eidheim, IDI NTNU
May 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Problem description

The problem for our research is motivated by the work done in the project Awesome Possum at the Norwegian Computing Center. The goal of Awesome Possum is to build an in-app passive authentication system for smartphones. They developed an authentication method that utilizes accelerometer data to authenticate users. However, they have encountered several challenges. One of the them is the need for long segments of data to be able to authenticate a user. For a passive authentication method to be useful in a real-world scenario, one must be able to authenticate a user with a relatively short time segment. This is due to the fact that an average smartphone session only last around 72 seconds [4].

Our goal in this project is to further explore the possibilities of passive authentication using accelerometer data from modern smartphones. We focus primarily on the possibilities of using short time segments of data. One goal is to build a model that may be versatile enough to authenticate a user in something close to a real-world scenario. In such a scenario, we cannot dictate the actions of the user. We aim to build a model that is considered versatile enough to be able to function regardless of the activity performed by the user. If we manage this, it should improve the user experience for the smartphone user.

Our plan is to build a baseline model able to perform passive authentication with a highly regular and complete dataset. We start with using a large portion of the data for training and a relatively small part of the data for testing and validation. In this case, the model is trained on every activity performed by the user. When the baseline model is established, we will use it to explore how different window lengths affect the accuracy. Thereafter we train the same model on a smaller subset of data. By reducing the amount of training data, we get a more irregular dataset. In this scenario the training data will not necessarily include all the activities present in the full dataset. Therefore, the model will need to generalize the learned representation to be able to classify future samples with previously unseen activities.

Abstract

Biometric authentication is researched due to the need for securing smartphones. As the number of smartphones in society increases, so do the possibilities that we have with our devices. These possibilities increase the potential damage that may be done by an attacker. Passive authentication methods could increase security without having a negative impact on the user experience. In this thesis, we propose an LSTM-based model that can passively authenticate smartphone users based on accelerometer data. We train and test our model on windows with a length of three seconds. This is far below the window lengths used in comparable studies such as the HMOG study [21] and in DeepAuth [7]. The proposed method achieves an AUC of 82% and an EER of 24.1% for the HMOG dataset [21]. Furthermore, we experiment with how different window lengths affect the accuracy of our model. The best results are achieved when using a length of three seconds. We also explore the performance of the model when trained on a small subset of the data. The experiment is performed to evaluate our model in a scenario that we consider to be more realistic. Our results for the experiment is an AUC of 75.84% and an EER of 29.28%. We also present insights gained from reproducing results from the scientific literature. We also present insights gained from reproducing results from the scientific literature.

Sammendrag

Biometrisk autentisering er et populært forskningsområde grunnet behovet for økt sikkerhet for smarttelefoner. Etterhvert som antallet smarttelefoner i samfunnet øker, øker også mulighetene vi har til å bruke dem. Det økte antallet av mulige handlinger vi kan utføre ved hjelp av smarttelefoner gjør at kravene til sikkerhet blir strengere. Passive autentiseringsmetoder kan potentielt bedre sikkerheten uten å ha en negative innvirkning på brukeropplevelsen. I denne bacheloroppgaven foreslår vi en LSTM-basert modell som kan gjennomføre passive autentisering av smarttelefonbrukere, kun ved hjelp av akselerometerdata. Vi trener og tester modellen med vinduer med en lengde på tre sekunder. Dette er en vinduslengde som er betydelig lavere enn det som er brukt i sammenlignbare studier [21, 7]. Den foreslåtte metoden oppnår en AUC på 82% og en EER på 24.1% på HMOG-datasettet [21]. Videre bruker vi modellen til å utforske hvordan vinduslengden påvirker treffsikkerheten. Vår modell fungerer best for en vinduslengde på tre sekunder. Vi utforsker også hvordan modellen blir påvirket av å bli trent på et mindre utsnitt av dataene som er tilgjengelig. Dette forsøket blir gjort for å utforske hvordan modellen fungerer i et scenario som vi antar å være mer realistisk. Resultatene av dette er en AUC på 75.84% og en EER på 29.28%. Vi presenterer også innsikt vi har fått ved å reprodusere resultater fra den vitenskapelige litteraturen.

Preface

In the last two years, I have become more and more interested in machine learning. Through subjects at NTNU and my part-time job, I have started to explore the domain to improve my knowledge. What motivates me about machine learning, is the opportunities that are uncovered when the technologies are used to improve existing systems. When trying to decide the topic of my bachelor thesis, I contacted Edvard to see if he had any ideas for a project. We then arranged a Skype meeting with Edvard, Bjarte, and I, to discuss the ideas at hand. As Bjarte explained about the work they do in Awesome Possum regarding authentication using accelerometer data, I became interested in exploring it myself. What motivated me, was to explore the domain of machine learning using temporal data, as well as the commercial potential of the project.

I started looking for a dataset suited for our problem. At the same time I performed a literature study to get an overview of the domain. Following the literature study, I started exploring different algorithms and preprocessing methods. This part of the study is fairly iterative, meaning that I tried multiple approaches and evaluated the results. If the results did not seem promising, I changed the approach and evaluated the results again.

Acknowledgements

First of all, I thank my supervisors Bjarte, Edvard, and Ole Christian. They have been available whenever I needed them for discussion and guidance. Their insights and constructive feedback have made a large positive impact on the quality of this thesis. I feel privileged to have such encouraging and engaged supervisors.

I'm grateful to my parents Sjur and Irmelin, my brother Heine, and my grandmother Edith for their love and support throughout my work.

At last, I thank my closest friends, who have supported and encouraged me during my work with this thesis. Your friendship means a lot!

Trondheim, May 2019

Sindre T. Nordal

Sindre Toft Nordal

Contents

1 Introduction 1

1.1 Hypothesis . . . 2

1.2 Summary of contributions . . . 2

1.3 Relevance . . . 3

2 Background 6

2.1 Smartphone authentication methods . . . 6

2.2 Human activity recognition . . . 7

2.3 Accelerometer . . . 8

2.4 Accuracy measurements . . . 8

2.5 Long short-term memory . . . 9

2.6 Concepts . . . 9

3 Dataset 10

3.1 Source of data . . . 10

3.2 Visualization . . . 11

3.3 Why HMOG? . . . 11

4 Approach 13

4.1 Data split . . . 13

4.2 Model . . . 14

4.3 Architecture . . . 15

4.4 Training the model . . . 18

5 Experiments 19

5.1 Experimental setup . . . 19

5.2 Baseline model . . . 20

5.3 Model trained with minimal data . . . 21

5.4 Window size . . . 22

5.5 Binary versus multi-class classification . . . 23

6 Related work **24**
6.1 Human activity recognition 24
6.2 Authentication 25

7 Discussion **27**
7.1 Results 27
7.2 Ethics 30

8 Conclusion **32**

9 Further work **33**
9.1 Performance 33
9.2 Commercialization 34

List of figures

3.1 One window of 2000 time steps of accelerometer data for two different users 11

4.1 The process of data splitting for one session 13
4.2 Architecture of the authentication framework 15
4.3 Representation of the three dimensional array containing the windows of accelerometer data. 17

5.1 AUC and EER in function of window length 22

List of tables

5.1 AUC of the baseline model with a window length of 3 seconds 21
5.2 AUC of the model trained with a subset of training data with a window length of 3 seconds 22
5.3 AUC and EER of the baseline model trained and tested on different window length 23

1 | Introduction

The increased availability and processing power of mobile devices has created many new opportunities. Smartphones are used for banking, to access confidential information, and to store private messages and images. Authentication methods for these devices include passcodes, passwords, swipe patterns, and biometric methods such as fingerprint recognition and facial recognition. These are examples of *one-time* authentication methods; they assume that once the phone is unlocked by the authentic user, it will only be used by the same user until the phone is re-locked. Applications such as Apple's App Store and most banking apps require explicit re-authentication to complete purchases or to be accessed. This means that they demand that the user authenticate himself before getting access to the application. If an unauthorized user gains access to an unlocked smartphone, he will have access to all information which do not require re-authentication. As information stored on modern smartphones can be highly sensitive, a hostile agent may inflict large damage.

Traditional authentication methods rely on information given by the user. They all require that the user perform a specific task. By demanding that the user perform a task, they negatively impact user experience. These methods also introduce latency. The combination of latency and negative impact on user experience has contributed to more than 28% of smartphone users not using passwords or PIN-codes on their devices [3].

Continuous passive authentication methods for smartphones have shown promise in multiple studies [17, 19, 21]. The studies show the possibilities of authenticating a user without the user actively performing a given task. Passive methods therefore overcome the drawbacks of the traditional methods. However, they are not ready to be used in a commercial setting, due to low accuracy or limited availability of data.

All modern smartphones possess accelerometers and gyroscopes that are continuously active. This has made it easy to access sensor data for research purposes. One of the most common fields of research using these sensors is human activity recognition (HAR). The goal of HAR is to identify the activity performed by the user given motion sensor data. Discoveries in HAR have led to studies involving identifying users based on the way they walk [26, 9]. These studies shows us that it is possible to authenticate users only using motion sensors in smartphones.

We propose an authentication method using only accelerometer data. We use the publicly available dataset HMOG [21]. This dataset contains data for 100 users performing three different tasks eight times. Four sessions of each task are completed while sitting, and four of the sessions are completed while walking. While the HMOG dataset is highly regular, it is not representative of a users behaviour in unctrolled environment. In a real-world authentication scenario, the user's activity is unknown. It is also likely that some users may walk and some may be seated. We perform an experiment where we train our model on a random subset of the dataset. By doing this, we try to increase the entropy of users' activities and by this come closer to a real-world scenario. Our theory is that a small subset of a user's set of data will not contain the same amount of the eight different activities. We consider this to lead us closer to a real-world scenario where all users do not perform the same activities. Given that our model still reaches an acceptable accuracy, we can say that our model may be versatile enough to be useful in a real-world scenario. Our model is not tested or deployed in a real authentication environment. All of the data used in our experiments are prerecorded. The experiments and results are therefore based on a simulated approach to authentication.

1.1 Hypothesis

For this thesis we have chosen the research hypothesis:

> It is possible to passively authenticate a smartphone user with an acceptable accuracy only using data from the built-in accelerometer in modern smartphones.

1.2 Summary of contributions

In this thesis, we focus on building a neural network able to perform passive user authentication using motion sensor readings from smartphones. We train and test the model on a publicly available dataset to make sure the results are replicable.

We use deep learning, as this approach has proven to be highly efficient to extract important features from raw data. Our model is based on a recurrent neural network (RNN), in the form of a long short-term memory (LSTM) network, which has shown good performance on temporal data.

Our contributions are:

1. An LSTM-based model able to perform passive authentication of users with an AUC 82% and an EER of 24.12% with a window-length of three seconds. The architecture of the model is described in chapter 4 and the experiments are described in chapter 5
2. An implementation of our proposed model for passive authentication using accelerometer data with time segments shoter than 15 seconds. In section 5.4 I present the results for our model for time segments of different lengths.
3. Insights from reproducing results from the scientific literature, including missing explanations and missing justification of choices. This is described in chapter 8.

1.3 Relevance

1.3.1 Conditions for commercial success

The most important metrics for evaluating an authentication method is security and usability. In our case, that means achieving a false acceptance rate (FAR) and false rejection rate (FRR) as low as possible. To be able to deploy our model commercially, we will need to be competitive to the solutions which are in use today. The FAR of solutions such as FaceID and TouchID is claimed to be lower than 1:400 000 [1].

To achieve a low FAR in itself is a relatively easy task. The challenge becomes increasingly difficult when trying to keep the false rejection rate (FRR) low at the same time. If a user is repeatedly falsely rejected by the authentication system, they are inclined to turn off the system. Therefore, it is essential to achieve both a low FAR and FRR.

One might conceivably also claim that our proposed model is commercially valuable even though it does not achieve a state of the art accuracy. Many smartphone users experience the latency caused by the use of traditional authentication methods as irritating. This irritation is one of many reasons leading people to not activating authentication methods on their smartphones. A study from Consumer Reports [5] discovered that 34% of Americans do not have any security measures active on their smartphones. For this group of people, a continuous or passive authentication method with a low FRR and a slightly higher than normal FAR would be a massive improvement of their device's security. A passive authentication method is not dependent of a user's direct input. Continuous methods are methods with repeatedly authenticate the user.

Another crucial factor for the commercial value of the continuous authentication method is the time needed between each re-authentication. Time spent on a smartphone or in a specific app varies greatly, but we can assume that the average use time is around 72 seconds [4]. Some apps, such as a banking app used to check account balance, may have an average of under 30 seconds of use time. For a continuous model to be commercially valuable, it would need to be able to re-authenticate the user multiple times within that interval.

1.3.2 Approaching a real-world scenario

The dataset used in this work is collected in a controlled environment. A consequence is that every user in the dataset performs the same set of activities. If we select a sufficiently large training set, the training of our model will include all possible activities in the dataset. In a real-world scenario, however, one could expect to encounter activities that are previously unknown to the model.

To come as close as we can to a real-world scenario, we explore the consequences of training the models on a small training set. We choose 50% of the windows for a user by random selection and use it to train the model. This causes the activities contained in the training data to be different from user to user and thus come closer to a real-world scenario. The subsampling of training data increases the difficulty of authenticating a user. By decreasing the amount of previously known patterns for a model, we rely on the model to detect more abstract patterns. If it does not identify these patterns, it will be unable to recognize the authentic user given that activities performed in the future are not identical to previous activities.

1.3.3 Structure of the thesis

The thesis can be divided into three parts. The first part includes this chapter and Chapter 2. Here, we introduce the research hypothesis, our contributions, and the conditions that needs to be full filled for commercial applicability. In the first part, we also introduce terms and concepts that are necessary to understand the following chapters.

The second part of our thesis consists of the chapters 3 to 5. In these chapters we explain the what dataset we use and the architecture of our model. We also explain the setup of our experiments and their results.

In the final part of our thesis, we focus on the discussion of our results. This part consist of the chapters 7 to 10. In Chapter 7 we describe related works and compare them to this thesis. This chapter also includes discussion of the choices made in related works compared to the ones we have made. In chapter 8 we discuss the results of our experiments, the architecture of our model, and the ethics regarding machine learning and authentication methods. The following chapter concludes our thesis. Here we discuss our results in relation to the problem description, the research hypothesis, and our contributions. Lastly, we discuss the possibilities of further work on biometric authentication using accelerometer data.

2 | Background

2.1 Smartphone authentication methods

2.1.1 One-time authentication methods

One-time authentication methods require that users identify themselves when they start using a smartphone. These methods only provide limited security. In the case of one-time methods, a smartphone that is unlocked will stay unlocked until it either locks itself after a certain time or is manually locked. Smartphones often let the user decide when to automatically lock their phone after it has been inactive for a certain amount of time. The other way of locking the smartphone is for the user to manually lock their phone. This gives people with bad intentions a relatively large window of opportunity to steal the unlocked phone and gain access to private information.

2.1.2 Continuous methods

To combat the easy access of unlocked smartphones, multiple papers [21, 20, 7] suggest continuous authentication methods. These include both biometric or non-biometric methods. The idea behind continuous authentication is to continuously re-authenticate the smartphone user at a given interval. As the average user only uses their phone for 70 seconds per session [4], the re-authentication interval would have to be rather short for it to be useful. Continuous authentication methods have proven to be promising, but have yet to be used by large smartphone manufacturers.

2.1.3 Passive methods

Passive authentication methods aim to authenticate a user without explicit input. The methods rely on information recorded by the smartphone. Variants of passive authentication can rely on motion sensors [20] or on use-patterns such as tap frequency and swipe patterns [21]. They improve the user experience by removing the latency introduced by other authentication methods such as PINs, facial patterns, and fingerprints.

2.1.4 Knowledge-based authentication

The most commonly used knowledge-based authentication methods are PINs and graphical patterns. All these require that the user remembers their choice of PIN, password, or pattern. If a user chooses a long and difficult PIN or password, it may secure their device in a good way. Unfortunately, many users choose short and simple PINs and passwords as this eases their use of the device. Studies have also discovered that users often use one password across multiple platforms, thus increasing the damage that may be caused by a thief or hacker.

2.1.5 Biometric authentication

We divide biometric authentication methods into two categories:*biometric-physiological* and *biometric-contextual*. Biometric-physiological is the methods that rely on the uniqueness of the physiological. This includes fingerprints, facial features, iris scans, ear shape, etc. Apple and Samsung are using such methods. Apple claims that there is only a 1 in 50000 chance that someone else's fingerprint will falsely unlock your iPhone [1]. Biometric-contextual methods rely on the uniqueness of the characteristics of the user. This includes their voice, gait, motion, hand-writing, etc.

2.2 Human activity recognition

Human Activity Recognition (HAR) is aiming to identify the activity performed by a given user at a given time. The technology in modern smartphones allows people to interact with their devices at almost all times of the day. This makes it possible for us to track users' activities based solely on their smartphones.

HAR has been deployed in wearables [27] and smartphones [6]. We now have smartwatches and smartphones which can identify the number of flights of stair climbed each day. They can identify if you are walking or running, if you are riding your bike or swimming, and much more. HAR systems rely on many different features. Most commonly the features include location data, accelerometer data, physiological signals, and environmental signals. Most systems would work while missing one of these features. Smartphones does a good job on HAR despite not recording physiological features.

2.3 Accelerometer

An accelerometer is a device that measures proper acceleration. We find it in every modern smartphone. The output of an accelerometer is the acceleration in the X, Y, and Z directions. Through the use of these features, a smartphone can, for example, rotate the screen from portrait to landscape, show current speed, and count steps. By plotting accelerometer readings, we can identify different events. We can, for example, determine that the smartphone is laying still with the screen up if the X and Y axes are zero and the Z axis is -1000. To detect steps, one can identify a forward motion combined with short spikes along one axis which appears as the feet hit the ground.

Accelerometer data has become increasingly popular in the case of HAR. Multiple works [16, 6] have found accurate solutions to this problem by only using accelerometer data. This has also reached commercial usage in devices such as smartwatches. Accelerometer data is often included as a part of larger biometric authentication systems when exploring continuous authentication. Given that accelerometer data is often sampled at a rate of around 100Hz, it is easy to produce continuous sequences of data.

2.4 Accuracy measurements

We use a set of standardized performance metrics to evaluate the performance of our model throughout this thesis. The following metrics are used:

- False acceptance rate (FAR) is the probability of classifying segments from a fraudulent user as an authentic user. The FAR is given by:
  $FAR = p(authentic|fraudulent)$
- False rejection rate (FRR) is the probability of classifying segments from an authentic user as a fraudulent user. The FRR is given by:
  $FRR = p(fraudulent|authentic)$
- Equal error rate (EER) is the percentage where the FRR is equal to the FAR. This is the overall metric used to evaluate the performance of our model.
- Receiver operating characteristic (ROC) curve is a curve used to represent the effectiveness of a binary classifier. The curve show True Acceptance Rate, i.e. 1-FRR as a function of the FAR.
- Area under the curve (AUC) is the area underneath the ROC curve.

2.5 Long short-term memory

Long short-term memory is an artificial recurrent neural network (RNN) architecture. It is used in the field of deep learning. RNNs have gained attraction due to the delivery of promising results in the field of sequence learning, time series prediction, and classification. They are being used for various experiments, including speech recognition [11], human activity recognition, and natural language processing [10]. However, RNNs do suffer from certain shortcomings. The most significant are vanishing gradient and exploding gradient. These problems result in the RNN being incapable of learning long-term dependencies.

LSTM is a version of an RNN which is intentionally designed to overcome the challenges of standard RNNs. The memory cells and gating mechanisms possessed by the LSTM make sure the layer retains the important information and forgets the information that it regards as less important. The information retained by the LSTM is the one that it can leverage to separate users. This trait makes the LSTM outperform other versions of RNNs when it comes to sequential datasets with larger time intervals. LSTMs have proven to perform well on classification tasks in the areas of HAR [25], speech classification, and text classification [23]. LSTMs normally outperforms other types of RNNs when the sequential data includes larger time intervals.

2.6 Concepts

This list is meant to serve as a dictionary for terms throughout the thesis.

Passive authentication – Authentication of a user without the user explicitly performing a given action. Normal approaches utilize keystroke patterns and behavioral analytics.

Continuous authentication – Continuously authenticating a user throughout the time where the device is in use.

Re-authentication – The user authenticating him or herself again or the user being authenticated again.

Explicit re-authentication – Asking the user to authenticate him or herself again. Normal when conduction in-app purchases or in banking applications.

Time segment – A portion of data that is recorded for a given amount of time.

Session – A recording of continuous usage of the device.

Window – A small subset of a session with a given length.

3 | Dataset

3.1 Source of data

The source of the dataset used in this work is HMOG [21]. It is publicly available along with a paper describing the collection methods and the contents of the dataset. The data was collected in a controlled environment from 100 unique individuals. They were asked to either read a document, write a text, or navigate on a map to a given location. Each activity was performed four times while sitting and four times while walking, by each user. This totals to 24 sessions per user. One session contains the accelerometer data recorded while the user performed a given task.

The researchers who collected the dataset developed their own data collection tool. They made an application for Android phones that each user used for the recordings. The features collected by the application were:

- Accelerometer
- Gyroscope
- Magnetometer
- Raw touch event
- Tap gesture
- Scale gesture
- Scroll gesture
- Fling gesture
- Key press on virtual keyboard

In this thesis we chose to focus on biometric authentication using accelerometer data. Therefore, we only use the recordings of this sensor. The smartphone used during the data collection has a 3D-accelerometer. This is the standard type of accelerometer in modern smartphones. The data is stored in CSV files categorized after the session and sensor they belong to.

3.2 Visualization

It can be helpful to visualize the data to get a better understanding of the data and differences between users. The following plot illustrates one window of accelerometer readings for two selected users.

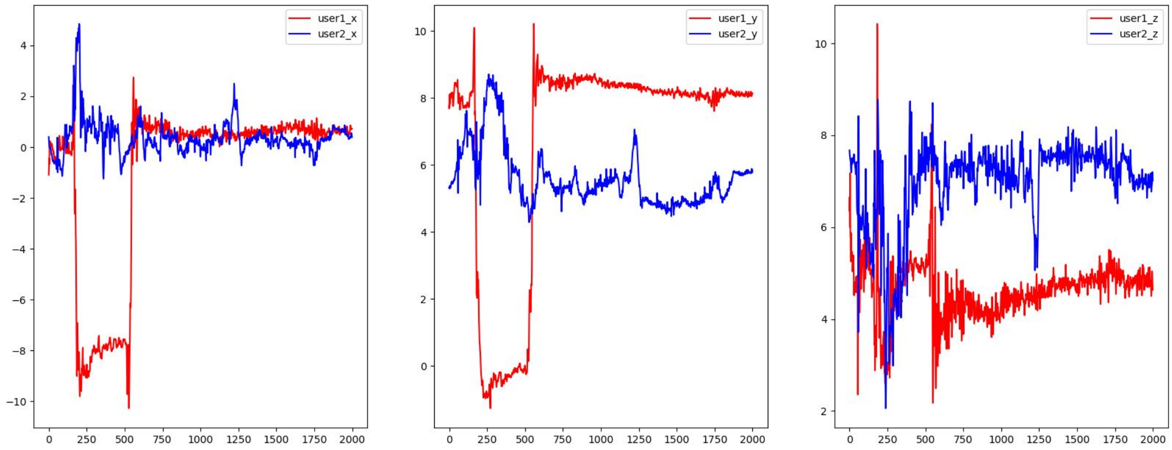

Fig. 3.1. One window of 2000 time steps of accelerometer data for two different users

From looking at the plot, we can easily distinguish the two users on all axes. However, we can also identify some similarities for the data on the x-axis. As we are able to identify the differences, it looks promising for our model to find some of the same differences.

3.3 Why HMOG?

We chose the HMOG-dataset for this thesis as we regard it as the most realistic dataset for a user authentication task. Most of the publicly available datasets containing accelerometer data are collected for, and used for, human activity recognition or gait recognition. A dataset designed for HAR would be challenging to leverage towards an authentication model. In these datasets, the accelerometer is most often placed in the front pocket of the subject or attached to their wrist or core [27, 6]. This means we would not have information about how the user uses their smartphone in the situation where we would like to authenticate them. We want to authenticate the user when the smartphone is used actively. Consequently, our model needs to be trained on data collected in this situation. HMOG is a dataset collected for developing an authentication model. The data was collected in a controlled environment. However, the controlled environment was a fairly close simulation of day-to-day smartphone usage. This makes the dataset a good match for our problem.

4 Approach

In this chapter we specify the architecture for our proposed model. This includes definitions and explanations of the preprocessing and an explanation of the architecture of our model. The values assigned to each variable and the specific model is described in the chapter 6.

4.1 Data split

We chose to split our dataset into training, testing, and validation data. The training data is used to train our model, the testing data is used to evaluate the trained model, and the validation data is used to validate the model on previously unseen data. In figure 4.2 we display the splitting operation. As seen on the figure we use the training and testing data for development and the validation data to validate our results.

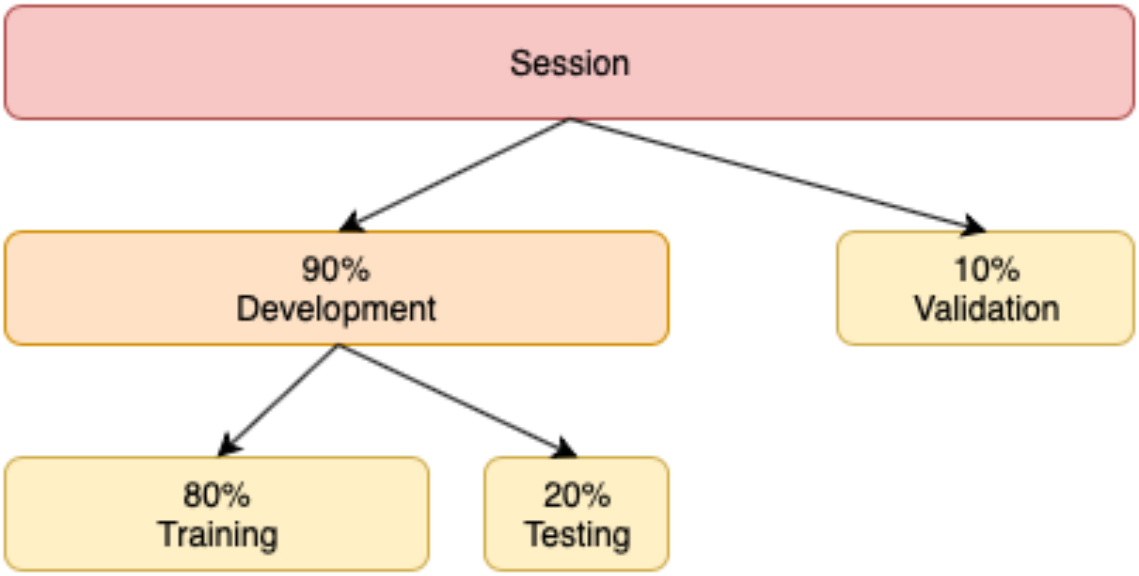

Fig. 4.1. The process of data splitting for one session

We train our model with our training data. During training, we evaluate the performance of our model with the training accuracy and loss. If we are not satisfied with the achieved accuracy, we tune the hyperparameters and train it again. When we achieve satisfying results, we test our model using the testing dataset. If our testing accuracy is substantially lower than the training accuracy, we know that we have overfitted the model. This means that our model has learned a representation of the training dataset that is not general enough to be applied to previously unseen samples. If this is the case, we have to re-train our model with fewer epochs or introduce a dropout layer. At this stage we introduce the risk of optimizing our model for the testing dataset. The test data is meant to be kept unseen from our model. If we detect overfitting and need to retrain our model, the testing data is no longer unseen.

We use our validation data to make sure that we have not optimized our data for a given testing dataset. The validation data serves as a replacement for a cross validation [14]. Cross validation is a way of validating testing results by training and testing the model multiple times using multiple different training and testing datasets. In our case, the cost of splitting the data multiple times is high, due to the large amount of data. The reason for cross validation is to make sure that the model is not optimized for one particular test set. As we only perform one test split, it is difficult to validate our testing accuracy. We handle this issue by holding out our validation data during training, testing and hyperparameter tuning the model. Our validation data is only used one time, and that is for calculating the final results of our experiments. If our model achieves a high accuracy on the validation data, we can be confident that our results are not the result of overfitting to the test set.

4.2 Model

The goal of the authentication model is to classify a user as authentic or fraudulent based on a window of accelerometer data. To perform the classification our model needs to be able to distinguish between the behavioural patterns of different users.

We define a window of data as a series of measurements for a period T. This can be modelled by a matrix $S = [s_1, s_2, \dots, s_n]$, where s_n is the accelerometer data at the timestep n. The total number of windows in the training data is given by $M = N_p + N_n$ where N_p represents the windows belonging to the authentic user and N_n is the windows belonging to a random selection of fraudulent users. We consider all users except the authentic user as fraudulent. The model is trained with a training set $X = [S_1, S_2, \dots, S_m]$ where S_m is a window of training data. The label set corresponding to the training data is given by $Y = [y_1, y_2, \dots, y_m]$ where $y^i \in \{positive, negative\}$. The label tells the model the correct class of each window of accelerometer data.

The model aims to detect behavioural patterns in each window that can be used to classify the window as either positive or negative.

4.3 Architecture

The proposed method consists of five different parts: preprocessing, windowing, LSTM, droupout, and the output. First of all, the accelerometer data is read from the CSV files and concatenated with their corresponding subjectID. The subjectID tells us the correct subject for the data. The subjectID later acts as the label for the session. After having loaded and formatted the positive data, we randomly sample and format negative sessions. Consequently, we pass the concatenated positive and negative data to the windowing layer, where we divide it into windows. The model used for the classification is an LSTM that aims to learn the representation of the behavioral patterns of each user. The output layer assigns the received window to either the positive or the negative class. An overview of the proposed architecture of the method can be seen in figure 4.2.

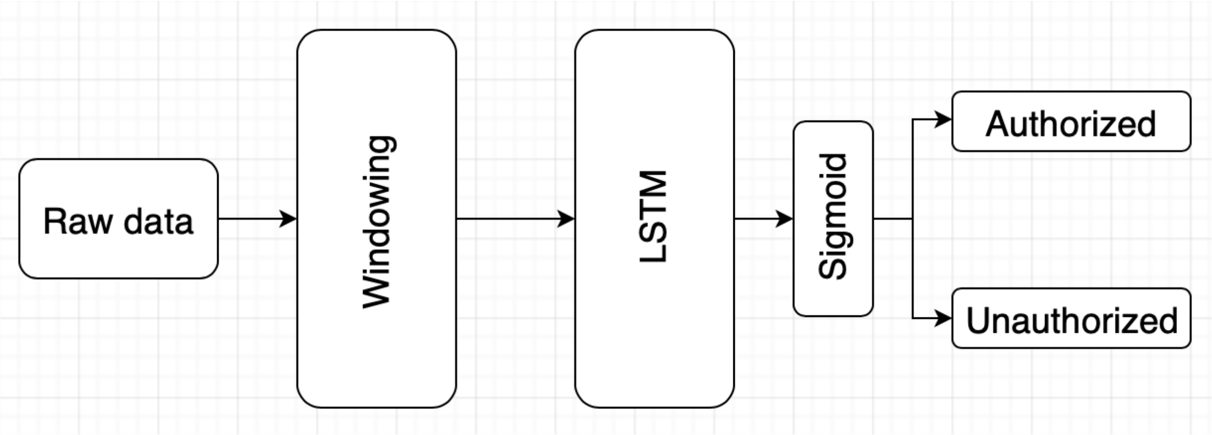

Fig. 4.2. Architecture of the authentication framework

4.3.1 Preprocessing

The accelerometer recordings in our dataset are sampled at a rate of 100Hz. As all of our data is collected with the same smartphone, the samples are already synchronized. When the accelerometer data is read from the file, it is concatenated with the corresponding subjectID. The vector for each time step is now on the form: $[x, y, z, \text{subjectID}]$. When the positive data is formatted, we load the negative sessions needed to train the model. After loading the negative samples, we check the ratio $r = \frac{\text{negative}}{\text{positive}}$. r is used to check for imbalance in the dataset. If a dataset is imbalanced it means that we have a large overweight of either positive or negative samples. This could affect the performance of our classifier by introducing unwanted bias towards one of the classes.

4.3.2 Windowing layer

When the accelerometer data from a session is loaded and formatted, we pass it to the windowing layer. This layer is used to break the larger sessions of data into smaller windows of data on which we train our model. The windowing layer makes it easier for the model to discover patterns hidden in smaller segments.

The windowing layer divides the sessions into windows of length T, where T is a given number of time steps. If $(\textit{segment length}) \bmod (\textit{length of window}) = 0$, the session in its entirety is divided into windows. If the equation is not equal to zero, the result gives us the number of overflowing timesteps. In the case of overflow, we will have to reduce the number of timesteps in order to fit it into windows. The number of overflowing timesteps is assigned to the variable O. We choose a random starting point $x \in \mathbf{N}$ where $0 \leq x \leq O$ so that we do not cut the same time steps for each session. If we always remove the first or last timesteps, we would risk removing important data from the start or end of the session. By randomizing which data to remove, we make sure that we do not consequently leave out the same part of the session.

After we divide a session into windows, all the windows are stacked to form a three-dimensional array. The order in which the windows are stacked is randomized. By randomizing the order, we force the model to evaluate each window individually. In a real-world authentication scenario we will often not have the opportunity to compare either the past or the future. Therefore, we believe this step brings us closer to a real-world scenario. When the array is constructed, we split it into X and Y, where X contains the accelerometer readings and Y contains the labels. Y is then reduced to a one-dimensional array as all time steps in a window necessarily has the same subjectID. X is kept as a three-dimensional array with the time steps on one axis, the accelerometer data on another axis, and windows on the third axis. Figure 4.3 illustrates the structure of X. Before leaving the windowing layer we perform a train-test split. The split divides X and Y into a given split between training data and testing data.

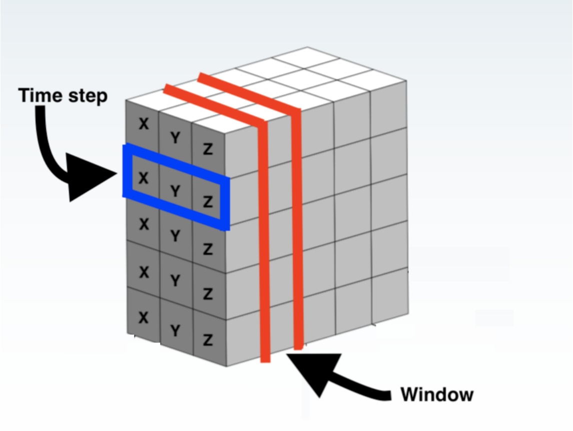

Fig. 4.3. Representation of the three dimensional array containing the windows of accelerometer data.

4.3.3 LSTM layer

The goal of our model is to be able to distinguish between different users by discovering different behavioral patterns. To discover the patterns we use an LSTM layer [12]. The LSTM layer is receiving the windows from the windowing layer and outputs a feature vector H. The LSTM layer applies the weights learned in training to each window.

4.3.4 Dropout layer

We include a dropout layer [22] between the LSTM layer and the output layer. The dropout layer randomly removes a number of connections between the LSTM layer and our output layer. We set a parameter p in the layer to a value between 0 and 1 to specify the number of connections to drop. The layer is included to prevent the model from overfitting.

4.3.5 Output layer

The output layer of our model is a fully connected layer with two output nodes. It receives the output of the LSTM and triggers one of its neurons corresponding to either a positive or negative class using the softmax activation function. The loss function used by this layer is the categorical cross-entropy function.

4.4 Training the model

We train the model using the whole ensemble of steps in section 4.3. The neural network is trained using back-propagation with respect to the categorical cross-entropy function chosen in the output layer. This means that the model will use the learned information in each epoch to try to minimize the loss calculated with the categorical cross-entropy. When training on a set of windows, the model aims to optimize the loss function using the optimizer called ADAM [13]. ADAM is an adaptive version of the stochastic gradient descent.

5 | Experiments

To investigate our research hypothesis we establish a baseline model. We use this model to experiment with different sized windows. Thereafter we experiment with training the same model on a smaller part of the total dataset.

5.1 Experimental setup

To train our model we use the IDUN cluster [2] provided by the HPC-Lab at NTNU. We use the NVIDIA Tesla V100 and the NVIDIA Tesla P100 for our computations. The cluster is using the SLURM [24] workload manager. We submit slurm-scripts to schedule our jobs, detailing the resources needed for the job and which Python scripts to run. The runtime-reports are saved in an output file. We save the trained models in the assigned working directory.

The size of the accelerometer data in the HMOG dataset is 12.02GB. Using a intel core i7 vPro 7th gen CPU, our baseline model uses approximately 7 seconds pr epoch. We train one model for each user, and run the training for 50 epochs. This results in a total CPU-time of nearly 10 hours for the baseline model experiment. When using the NVIDIA Tesla P100, each epoch is recorded at 0.4 seconds. This results in a total training time of 33 minutes for the exact same experiment. The latest job we ran on the IDUN cluster used a total 43 minutes. This implies that our model used approximately 10 minutes for loading and preprocessing the data.

The cluster is used by multiple students, employees, and researchers at NTNU. We have experienced our jobs being queued for more than 10 hours before gaining access to the required resources. To maximize the potential of the cluster, it is important that each user does not use more resources than necessary. As a consequence we choose to only run our jobs on one GPU. This choice also gets our job a higher priority in the queue. The higher priority is achieved if only one GPU gets available and the other jobs in the queue require multiple. While we did not use multiple GPUs, we did explore the possibilities of running multiple tasks on one node. In this case, the size of our dataset caused the memory of the GPU to overflow.

5.2 Baseline model

The purpose of building a baseline is to have a model we can compare with the rest of our results. The model also shows the best accuracy we have achieved for our passive authentication. We train each binary model with data from the authentic user and data from randomly selected non-authentic users. 80% of the development data from the authentic user represents the positive samples in the training data. After loading the positive samples, we divide them into windows with a length of three seconds. We use a loop ranging from 0 to the number of windows in the positive data, to sample the negative windows. For each iteration we select a random non-authentic user, and load a window of training data from this subject. The subjects from who we load the negative samples are selected by random choice. This serves to reduce the risk of introducing human bias to our model. If we select a few users to use as negative samples, our model would be biased towards these users. This means that our model would have an easier time classifying these specific users, but would have a harder time classifying previously unseen users. As we did for the positive samples, we split the negative samples to 80% training and 20% testing.

The model is trained with a window length of 300, which is equivalent to 3 seconds. We fix the relationship between the positive and negative samples to $r = 1$. This is done to make sure that our model is trained on a balanced dataset. An imbalanced dataset may introduce unwanted bias towards one of our classes. If we use all of the available training data, we would have a ratio of 1:99 between positive and negative windows. This results in our model potentially achieving a 99% accuracy by classifying all samples as negative. The goal of our model is to reduce the loss function. It is therefore inclined to exploit the imbalance by being more sensitive to the class with the most windows. We chose to handle the imbalance by subsampling the negative samples. The subsampling is performed by extracting the same amount of negative windows as the amount of positive windows. The model is trained with the default parameters for the ADAM optimizer in Keras. This includes a learning rate of 0.001. We set the batch size to 512 and train our model on 50 epochs. The number of hidden units in our LSTM layer is set to 25. The setup is inspired by the DeepAuth study [7].

We use data from the validation split for each user to evaluate the performance of our models. The validation data is a portion of data that is kept away from our model during both training and testing. As we have trained one model for each user, we loop through all the models to validate our results. As done by Amini et al. [20], we validate our model using the validation samples from the authentic user and the validation samples for one non-authentic user. The result from our validation can be seen in table 5.1. Our results show the area under the curve (AUC) and equal error rate (EER) for our model using a window length of three seconds.

Window length	**AUC**	**EER**
3 seconds	82.00%	24.12%

Tab. 5.1. AUC of the baseline model with a window length of 3 seconds

The results shows us that we are able to passively authenticate a user with an AUC of 82% and an EER of 24.12% using a window length of 3 seconds. The results of our experiments are around 10% below the comparable results found in comparable scientific literature [21, 7, 20].

5.3 Model trained with minimal data

The goal of this experiment is to evaluate our model on a dataset that we consider closer to a real-world scenario. In this experiment we extract a random subset containing 50% of the windows to use for the training of our model. The subset will not contain the same set of activities for each user. We therefore argue that it is closer to a real-world scenario where every user performs different activities. We train our model on 50% of the positive data available, test the model on 10% and use the remaining 40% for validation. The negative samples are chosen by random selection, as for the baseline model, and are split in the same percentages as the positive data.

We choose to train our model on a window length of 3 seconds. Our model is trained with the ratio $r = 1$ between positive and negative samples to avoid imbalance in the training data. We use the same choice of variables for our neural network as we use for our baseline model. This is described in section 5.2. By keeping the setup the same, and only changing the percentages of our data split, we make sure that the difference in results can be traced back to that operation.

We use the validation data for each user as positive samples for their respective models. The negative windows used for validation are selected at random from another user in the dataset. The result for the validation can be seen in table 5.2. Our results show the AUC and EER for the model using a window length of three seconds.

Window length	AUC	EER
3 seconds	75.84%	29.28%

Tab. 5.2. AUC of the model trained with a subset of training data with a window length of 3 seconds

The results from the validation show that the performance decrease a small amount when training on a subset. We do regard the results as fairly promising as they did not decrease more. The difference in EER between our baseline model and this model is 5.16.

5.4 Window size

To further investigate the potential of using short window lengths of data, we train our model on multiple different settings. Our goal is to explore whether the window length affects the accuracy of the baseline model. As mentioned in the relevance section, an important aspect for commercial applicability is the time needed to authenticate a user. This experiment will provide us with relevant insight into this challenge of commercialization.

We experimented with windows of sizes [100, 300, 500, 1500] – corresponding to 1, 3, 5, and 15 seconds. We ran our baseline model with the given window sizes to compare the achieved accuracy. The setup for our model is mostly the same as for the experiment with the baseline model. The only parameter that we vary is the window length. The AUC and EER achieved by our baseline model with varying window lengths are displayed in figure 5.1. The results are also displayed in table 5.3

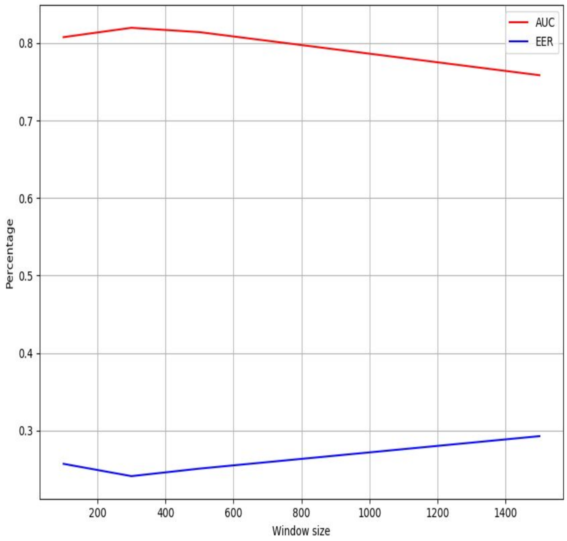

Fig. 5.1. AUC and EER in function of window length

Window length	AUC	EER
1 second	80.75%	25.71%
3 seconds	82.00%	24.12%
5 seconds	81.40%	25.09%
15 seconds	75.84%	29.28%

Tab. 5.3. AUC and EER of the baseline model trained and tested on different window length

The results show that a window length of three seconds results in the best accuracy. We also see a trend where a shorter window length than three seconds and a longer window length than three seconds results in a decrease of accuracy.

5.5 Binary versus multi-class classification

At the beginning of our project, we trained a model with N output nodes given N unique users. We based the model on the same architecture as the one we propose in this thesis. The difference between the multiclass classification model and the binary classification models is the output layer. By using a multiclass classification model, we only need one model to classify all the users.

Our experiments with the multiclass classification model gave us an AUC of approximately 90% for two users. As we increased the number of users, the performance of the model declined. For ten users, our model only achieved an AUC of 45%. The main drawback of using multiple binary classifiers for a multiclass problem is the risk of getting multiple positive results. This means that multiple binary classifiers predict that a given sample belongs to their class. In our case, this is not a problem. For authentication of smartphone users, we try to authenticate the owner of the smartphone; we know which binary classifier to use and therefore avoid this problem.

The results in this experiment made us move away from a multiclass approach and over to a binary classification approach.

6 | Related work

There are more than three billion smartphone users in the world. Each smartphone has built-in sensors that capture data almost continuously. The easy access and availability of the data collected by these sensors has led to extensive studies across multiple fields of research. Human activity recognition and continuous user authentication are some of these research areas.

6.1 Human activity recognition

Kwapisz et al. [16] performed HAR on a dataset of 29 volunteers performing six different activities. All subjects carried the phone in their front pocket while performing the activities. Further, they reviewed the performance of a logistic regression and a multilayer perceptron for the task. The multilayer perceptron achieved an accuracy of 91.7% overall for the given dataset. Although the overall performance is reasonable, we see that the model only achieved an accuracy of 44.3% on the activity of walking down stairs. The reason for the low accuracy is that the model had difficulties with the distinction between walking up and down stairs. This illustrates that the multilayer perceptron may have difficulties when there are only subtle differences between samples. Even though the model used by Kwapisz et al. is not the same as we use, we can identify some similar problems. During testing, we have discovered that certain users are more similar in behavioural pattern than others. This causes low accuracy for certain users, just like Kwapsiz et al. experience for the activity of walking down stairs in their study. This challenge made us move from a single multiclass model towards multiple binary models.

Alsheikh et al. [6] propose a deep learning approach for HAR. With a deep learning architecture, they outperform shallow methods. They manage to achieve an accuracy of 98.23% on the same dataset used by Kwapisz et al. in the study discussed above. The same model also achieved better than previously achieved results on other popular datasets. One of the advantages with deep models is the fact that one usually does not handcraft features. For shallow models, it is typical to create features with the raw data and then pass it to the model. With deep models, we rely on the neural network to learn the important behavioural patterns from the raw sensor data.

6.2 Authentication

Sitová et al. introduce the HMOG dataset [21], on which they perform continuous user authentication using multiple sensors from smartphones. By using all the features available in the HMOG dataset, they achieved an EER of 8.65%. They achieve better results than our method, but they also use a much larger specter of sensors. By leveraging all the smartphones sensors they have to accept an increased cost of processing. There seems to be a trade off between accuracy and required computational power. Computational power is getting less expensive and more easily available. We therefore find the increased computational burden reasonable, as it seems to provide them with improved security.

Centeno et al. [20] achieved an EER of 2.2% with an autoencoder. They tested their model both on the HMOG dataset [21] and on a real-world dataset. They do not mention whether they have one model for each user or whether they have one model for all users. They do however mention that they train the model for the positive samples and test it with samples from a randomly selected negative user. Given the nature of autoencoders and that statement, we assume that they trained one autoencoder per user and framed it as an anomaly detection problem. That means that they have chosen a slightly different approach than our proposed method. They evaluate their model with a window length of 20 seconds. Centeno et al. evaluate the performance of their model by using samples from only one negative user. We have chosen the same approach to be able to compare our results with their study and to DeepAuth [7]. It should however be discussed whether this is an accurate way of evaluating the performance. As the selection is random, there is no way of introducing human bias. There are however a risk of the selected user not being representative for all the other negative users in the dataset. This could lead to either a worse or better result for the model.

Amini et al. [7] have developed a framework named DeepAuth. They gathered sensor data from users of the Target mobile application to perform re-authentication. By using data from the accelerometer and the gyroscope they achieved an AUC of 96.70% with a re-authentication time of 20 seconds. This study is very much like the one conducted in this paper. The main differences between our study and theirs are the use of different datasets and the different preprocessing steps. The proposed methods and models are nearly identical. They did something interesting in their preprocessing where they conducted a fast Fourier transform (FFT) to map accelerometer and gyroscope from the time domain to the frequency domain. Their scatter plots show us that the distance between users increased by a significant amount by mapping the data to the frequency domain.

Amini et al. have, however, made some choices that they have not justified. They have chosen the same approach as us when it comes to creating one binary classification model for each user. However, they do not address the problems of choosing the negative samples in their work. As it is not clear how the testing was done, it is difficult to precisely judge the merit of the reported results. As they do not state their choice in this case, it is difficult to say how they selected the negative users for testing.

7 | Discussion

This chapter is divided into two sections, “Results” and “Ethics”. In the “Results” section, we discuss the results of the experiments and potential reasons for our accuracy being lower than in comparable studies. The setup for the baseline model remained the same through all the experiments. Discussion regarding the choice of hyperparameters and the structure of the model is therefore only discussed in section 9.1.1. In the “Results” section we discuss ethical problems that come with exploring authentication algorithms and machine learning in general.

7.1 Results

7.1.1 Baseline model

We regard the results achieved by our model as promising. Our baseline model shows that we are able to leverage LSTM networks to perform passive authentication of smartphone users. The results we achieved are $10 - 15\%$ below the results achieved by comparable studies [21, 20, 7]. This can be due to multiple reasons. One of the reasons that affects the performance is the choice of small time segments. From our experiment with window sizes, we could not see that an increased window length had a positive affect on accuracy. The reasons for that may have been that we kept the same architecture of our model for all window lengths. By increasing the number of hidden units in the LSTM layer, we would also increase the the capability of the layer to retain information. This could potentially have shown better results for windows of a length around 20 seconds as used in DeepAuth [7]. Another aspect that may affect our results, is the choice of approach. We see that Centeno et al. [20] has managed to achieve a higher accuracy than us. They chose to attack the problem of passive authentication as an anomaly detection problem. In hindsight this is an interesting idea that should be investigated further.

As mentioned in related work, an interesting topic of discussion is the selection of users for the testing of the model. In our study we test, and validate our testing results, using the positive samples from the owner of the model, and negative samples from one other user. We chose this approach as it was described by Centeno et al. [20]. By choosing the same approach we increase the value of comparing our results. In DeepAuth [7], they mention how they choose negative samples for training, but not for testing. This means that we do not know how they select users for the testing of their model. If they test their model using samples from the sames users that were used for training, they could potentially improve their results. We do perform the selection of the negative user for testing and validation at random. By using our approach with random selection of the negative samples in two different operations, we are sure that we do not use the same users for training and testing. The choice of selecting negative samples from one user, instead of using all the negative users, do introduce the risk of unwanted bias. We can not be sure that the selected user is representative of the rest of the dataset. The results are calculated as the mean score of all 100 models. This implies that we select 100 negative users, one for each model. The fact that our results are the mean score of our models does potentially reduce the impact of an unfair result of one model. This increases our confidence that our results are justified.

Another choice that may effect the performance of our model is the choice of randomising the order of the windows. In section 4.3.2 we argue this makes it harder for our model to learn the behavioural patterns in the data. The reason is that we remove the temporal dependencies between the windows. This forces our model to detect the behavioural patterns from each window separately. In a real-world scenario we depend on the ability to authenticate a user in a short amount of time. We argue that removing the potential of discovering inter-window dependencies helps us come closer to this scenario.

Due to the cost of computational power, it is important to discuss whether we could have obtained the same results with a simpler approach. In the HMOG study [21], they achieved a better result than us using an SVM. They did however use all the available sensors in the dataset. We chose to use an LSTM model, as the accelerometer data is structured as time series. The LSTM does increase the complexity of our model in comparison to an SVM. Based on the results of the studies of Zhao et al. [25] and Amini et al. [7], we think that the improved results justify the increased complexity. Both of these studies show that the LSTM outperforms shallower methods.

7.1.2 Window size

The experiment shows that the best results are achieved with a window length of three seconds. Any window length shorter or longer than that decreases the accuracy. We used the baseline model with the same configurations for all window lengths. Our initial theory was that a longer window length would increase the accuracy. The reasoning behind the theory was grounded in challenges met in the Awesome Possum project and in the use on window lengths around 20 seconds in the literature [21, 7]. A potential reason for our results not showing higher accuracy for longer window lengths may be the use of the same model configurations throughout the experiment. An increased number of hidden units in the LSTM layer could improve the potential of the model to retain information for a longer time period.

The goal of the experiment was to explore how window lengths affected the accuracy and to see if it was possible to authenticate a user with a relatively short window length. Our results show that an increased length does not necessarily imply that accuracy will be higher. This is corresponding well to the results achieved by Banos et al. when they studied the impact of window length for HAR [8].

7.1.3 Model trained with minimal data

The goal of the experiment was to evaluate how our model performed in a situation that we considered to be closer to a real-world scenario. To approach this scenario we trained another model on a subset of the training set. This model achieved an AUC of 75.84% and an EER of 29.28% for a window length of three seconds. This is not significantly worse than our baseline model. These results suggest that the model is fairly versatile and is able to distinguish users even with limited training data.

7.1.4 Model architecture

Our model architecture follows a fairly standard architecture. We primarily follow the same preprocessing steps as used in DeepAuth [7] and by Centento et al. [20]. The setup with one LSTM layer, a dropout layer, and softmax output layer can also be found in several works [25, 7]. We did some initial experiments with more than one LSTM layer. The results obtained by the models were not better than the model with one LSTM layer. It did, however, increase the training time for the model by a significant amount. Due to that, we chose to abandon the multiple layers.

The ADAM optimizer [13] is also found in DeepAuth [7]. We did not experiment with different learning rates. The learning rate specifies how fast the weights of our model changes. If the learning rate is set too high, we will risk not hitting the global minimum for the loss. This will result in the training loss being volatile and moving up and down. If the learning rate is too low we will risk getting stuck on a local minimum. During the training of our model, the loss was mostly strictly decreasing. We therefore chose to keep the learning rate at 0.001, which is the default for ADAM.

7.1.5 Simulation of authentication

We used a prerecorded dataset for our experiments. This implies that we did not get to test our model in a real authentication scenario. In an authentication situation the model would have to be trained on the history of a user and be able to classify newly recorded samples. Our approach is based on training on a set of data, and then testing our model with previously unseen data. The similarities between our experimental setup and a real scenario are many. Our setup also resembles the one used by Amini et al. [7]. Their study were deployed in a real scenario using customers from the Target application on Android phones. The similarities in the approach makes us confident that our results are not misleading. It is, however, hard to quantify how close to a realistic scenario we are, without testing our model in such a scenario.

7.2 Ethics

Accelerometer data is continuously sampled by modern smartphones. Unlike sensors such as microphones and smartphones, accelerometers are widely regarded as not being privacy-intrusive. As a consequence, most smartphone operating systems do not prevent third-parties from accessing accelerometer data. Kroger et al. [15] studied how accelerometer data can be exploited in a way that harm privacy. In their work they mention how studies show that the data can be used to detect PINs, analyze user activities, find user location and identify the age, gender and emotion of a user. A part of their conclusion is a call for action to review the privacy policies regarding accelerometer data. Our proposed method show that we are able to identify users with an acceptable accuracy. The fact that a user can be identified by data that is not secured from third parties raises some concerns. It means that fraudulent websites and applications may obtain information that can be used to identify users with our model.

Authentication methods play an important role in the life of smartphone users. It is necessary that the methods are robust enough to secure the sensitive information stored on the device. As computer engineers we have to perform risk analysis regarding our products. In this work, that means to evaluate the potential threats that may be introduced by the deployment of an accelerometer-based authentication method. As mentioned in the introduction, 34% of smartphone users in the US do not protect their smartphone using authentication methods. A passive method that removes latency and does not negatively affect user experience could potentially result in more users activating authentication methods. The method would still have to be as secure as existing methods such as PINs and fingerprints to not introduce a false sense of security. If users believe that their smartphones are more secure than they actually are, it could result in a more hazardous user behaviour. As computer engineers and developers of algorithms, it is important that we do not deploy a product that could have a potentially negative impact on the security of the devices.

An interesting discussion, which concerns the ethics of machine learning, is the bias present in the data. This means that we have to make sure that our model is developed with a dataset that is representative of the users that may use the model. Nagpal et al. [18] discovered that facial recognition algorithms often do contain prejudice towards users of a certain age or race. The same prejudice may be present in accelerometer data. In our thesis, we use the data collected in the HMOG study [21]. The authors of this study do not specify the age, race or sex of the participants in their study. Therefore, we have can not know whether or not they are representative for the society. We can as a consequence not be certain that our model is not prejudiced.

8 | Conclusion

We propose a machine-learning model that shows promising results for passive authentication by exploiting users' behavioral patterns. Throughout our experiments we obtained an AUC of 82% and an EER of 24.12% for our baseline model, which is comparable to the literature. Our results validate the hypothesis by showing us that it is possible to passively authenticate a smartphone user using only the accelerometer data. The accuracy is acceptable for a proof of concept, but is not high enough to be commercially valuable.

Our experiments with different window lengths show us how they affect the accuracy scores. We achieve the best results for a window length of three seconds. They also show the potential of using a window length of one second without a large decrease in accuracy. The experiments also show that, for our model a window length shorter or longer than three seconds will result in a decrease in accuracy.

The baseline model trained on a subsampled dataset obtained an AUC of 75.84% and an EER of 29.28%. This shows us that the model is able to distinguish between users when trained on a small subset of the data. The small subset of data is extracted randomly and increases the differences between the users. This is considered to be closer to a real-world scenario. Our conclusion on this experiment is that our model seems to be versatile enough to function in a more realistic environment.

Furthermore, we have gained insights in the scientific literature and discovered missing explanations and justifications. The missing explanations mean that it is unclear whether the reported accuracy is correct, or if it should be somewhat lower due to the nature of the experiment.

Our results answer well to the given problem description. We have explored the possibilities of passive authentication with short window lengths, as well as how the model performed in a closer to real-world scenario. The results are promising both regarding short window lengths and for the experiment with a subsampled training set.

9 Further work

This chapter describes our thoughts about further work into biometric authentication using accelerometer data. We chose to divide this chapter into the sections “Performance” and “Commercialization”.

9.1 Performance

We mentioned in our conclusion that our proposed methods can serve as a proof of concept. To be able to be used as an authentication algorithm in real scenarios, the accuracy would have to be improved. There are still potential ways to do this that we have not explored. One idea is to increase the number of LSTM layers to three or more. We tried with two, without achieving satisfying results. To stack more LSTM layers is an interesting way of trying to extract more performance of the model.

We also see potential in the preprocessing. In the early stages of our project, we explored the possible benefits of normalizing the data. This did not improve our results when using a min-max scaler between -1 and 1. It would be interesting to explore other scaling methods. Centeno et al. [20] and DeepAuth [7] have chosen to include overlapping windows in their preprocessing. This means that they shift windows by a factor T so that new windows include some timesteps from previous windows. Neither of these studies explain why they do this. It would be interesting to investigate whether this operation improves the results of our model.

In DeepAuth [7], the authors proposes a preprocessing step including a fast Fourier transform. When plotting the feature space, their results show increased separation of users by using this approach. It would therefore be interesting to see if this approach would increase the performance of our proposed approach.

As described in chapter 7, the choice of the negative samples for each model is considered important for the evaluation of performance. Ideally, we would use all the negative samples available in our dataset. Due to the problems that comes with imbalanced datasets, we choose to subsample the negative samples. Another tactic for dealing with imbalanced datasets is to over-sample the positive samples. Due to the computational burden, we chose the first method. Given the time and resources it would be interesting to compare the two approaches for dealing with the imbalance in the dataset. For the validation of our results, we select data from the authentic user, and only one negative user. To further explore the performance of our model, it would be interesting to experiment with including data from more non-authentic users during validation.

Another approach which has not been tested is an anomaly-detection-based method. Anomaly detection approaches have been effective in cases like credit card fraud detection and in predictive maintenance. An anomaly detection approach will also be based upon one binary model for each subject. The main difference is that one would probably use an unsupervised model.

9.2 Commercialization

One of our goals in this paper is to create a model which can be useful in a real-world scenario. In our thesis we simulate an authentication scenario. The accuracy of our model is at this stage not good enough to be commercially useful. To improve the performance of the model should therefore be considered the first step towards commercialization.

There are multiple aspects that we would like to explore further. First of all we would like to test our model in a real-world scenario. This means to deploy our model in a system where it can operate on data from users in an uncontrolled environment. In a real-world deployment one would get interesting feedback on whether our model actually performs as well in a scenario where we have no control over the behavioural pattern of the user as we assume in this paper.

As previously mentioned there are many conditions that needs to be fulfilled for commercial success of a passive authentication model. One of the conditions is usability. It would therefore be interesting to perform a test in a controlled environment where our model was tested by users. In a controlled environment, we would be able observe and dictate the behaviour of the user. This would allow us to test the user experience and evaluate it through different experiments. The user tests could provide valuable feedback about what users think about a system of this type.

References

[1] Face id security. https://www.apple.com/business/site/docs/FaceID_Security_Guide.pdf. Accessed: 09.04.2019.

[2] Idun cluster. https://www.hpc.ntnu.no/display/hpc/Idun+Cluster. Accessed: 15.05.2019.

[3] Many smartphone owners don't take steps to secure their devices. https://www.pewresearch.org/fact-tank/2017/03/15/many-smartphone-owners-dont-take-steps-to-secure-their-devices/. Accessed: 10.04.2019.

[4] Mobile user experience: Limitations and strengths. https://www.nngroup.com/articles/mobile-ux/. Accessed: 15.05.2019.

[5] Smart phone thefts rose to 3.1 million in 2013. https://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.html. Accessed: 05.05.2019.

[6] Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee Pink Tan. Deep activity recognition models with triaxial accelerometers. 11 2015.

[7] Sara Amini, Vahid Noroozi, Amit Pande, Satyajit Gupte, Philip Yu, and Chris Kanich. Deepauth: A framework for continuous user re-authentication in mobile apps. pages 2027–2035, 10 2018.

[8] Oresti Baños, Juan Manuel Gálvez, Miguel Damas, Héctor Pomares, and Ignacio Rojas. Window size impact in human activity recognition. In *Sensors*, 2014.

[9] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing*, 16(5):563–580, Jun 2012.

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger

Schwenk, and Y Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 06 2014.

[11] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 38, 03 2013.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[14] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14, 03 2001.

[15] Jacob Leon Kröger, Philip Raschke, and Towhidur Rahman Bhuiyan. Privacy implications of accelerometer data: A review of possible inferences. In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, ICCSP '19, pages 81–87, New York, NY, USA, 2019. ACM.

[16] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, March 2011.

[17] Chien-Cheng Lin, Chin-Chun Chang, and Deron Liang. A novel non-intrusive user authentication method based on touchscreen of smartphones. pages 212–216, 07 2013.

[18] Shruti Nagpal, Maneet Singh, Richa Singh, Mayank Vatsa, and Nalini K. Ratha. Deep learning for face recognition: Pride or prejudiced? *CoRR*, abs/1904.01219, 2019.

[19] Natalia Neverova, Christian Wolf, Griffin Lacey, Lex Fridman, Deepak Chandra, Brandon Barbello, and Graham Taylor. Learning human identity from motion patterns. *IEEE Access*, 4, 11 2015.

[20] Mario Parreno-Centeno, Aad van Moorsel, and Stefano Castruccio. Smartphone continuous authentication using deep learning autoencoders. pages 147–1478, 08 2017.

[21] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, May 2016.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan

Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

[23] Dong Wu and Mingmin Chi. Long short-term memory with quadratic connections in recursive neural networks for representing compositional semantics. *IEEE Access*, PP:1–1, 01 2017.

[24] Andy B. Yoo, Morris A. Jette, and Mark Grondona. Slurm: Simple linux utility for resource management. In Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 44–60, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[25] Zhao Yu, Yang Rennong, Chevalier Guillaume, and Gong Maoguo. Deep residual bidir-lstm for human activity recognition using wearable sensors. 08 2017.

[26] Jaeseok Yun. User identification using gait patterns on ubifloorii. *Sensors (Basel, Switzerland)*, 11:2611–39, 12 2011.

[27] M. Zubair, K. Song, and C. Yoon. Human activity recognition using wearable accelerometer sensors. In *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–5, Oct 2016.

NTNU

Norwegian University of
Science and Technology