Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Fraud Prevention and Detection in Mobile Payments

Master Thesis

Cedric Waldburger

wcedric@ee.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zurich

Supervisors:
Christian Decker
Conor Wogan (SumUp)
Prof. Dr. Roger Wattenhofer

April 10, 2013

Acknowledgements

I would like to thank my advisors Prof. Roger Wattenhofer and Christian Deck from the Institute for distributed computing at ETH Zurich for their help and support in the last months. I also wish to thank Conor Wogan, Matti Biskup and the whole Software Development Team at SumUp for their continuous support and help.

Abstract

Mobile payments have seen a lot of traction recently and a wide variety of companies have emerged around the opportunity of payments that are processed from a mobile device. Our work was conducted in collaboration with a company that allows merchants to process credit card transactions on their smartphone by using an app and hardware dongle.

The innovative technology has opened up many new benefitial opportunities for both, merchant and clients. But it has also openend new opportunities for fraud in the area of credit card transactions, an area which has always been prone to fraud. Although the industry is highly regulated, the advances in technology call for new anti-fraud measures.

We identify two separate but complementary methods against credit card fraud: **preventive**, in the form of an automatic, more in-depth check of all users during the sign-up process against a database of individuals with higher risk status, and **reactive** by providing mechanisms to check every transaction's signature in realtime against the cardholer's previous signatures.

Our work has shown that the optimization of the check during sign-up is able to reduce the number of false positives by (TODO: XX percent). This results in up to (TODO: XX hours) less manual work by operations team each week.

We found that the characteristics of a signature captured by finger on a mobile device are much less stable than those of signatures captured with a pen. Therefore, we propose to use a feedback loop to improve the reliabilty of the signature verification algorithm and reduce the EER of our algorithm which fuses the score of a DTW algorithm, an HMM and a score base on globabl features.

Keywords: Fraud Detection, Signature Verification, Fraud Detection, Mobile Payments, Dynamic Time Warping, Hidden Markov Models

Contents

Chapter 1

Introduction

As Smartphones become increasingly popular, they're being used in more and more industries and fields. An application that has seen a lot of traction recently is mobile payments. The usage of the smartphone in this field includes the utilization as an electronic wallet, a virtual bank account and as a payment processing terminal to accept credit cards and other payment methods. This work focuses on an application of the latter. The density of smartphones per capita and the computational power of current phones have reached a level where it became practical to use phones electronic cash registers. iZettle[1], Square[2], SumUp[3] are just some of the companies which make use of this situation and are enabling merchants to receive payments via their smartphone. This thesis was written in collaboration with SumUp, a mobile payment company located in Berlin.

Financial businesses are subject to fraud more than average due to the direct financial gain for fraudsters. On mobile devices, it becomes even more important to protect the system from Fraud — especially as the mobile devices can be lost or stolen. While some security measures are imposed by the financial authorities, those rules were made for traditional financial institutions. The new mobile, connceted environment not only opens up a lot of beneficial opportunities but also more attack vectors for fraudsters. Therefore it is important to not only respect given guidelines but proactively look for ways to prevent possible attacks.

We focused our work on two separate, but complementary approaches to reduce the fraud risk for a mobile payment company like SumUp. Thanks to implementing an in-depth check of new users during the sign-up process, we preventively reduced the risk of signing up a high risk account. After screening a new user, suspicious applications are flagged for manual inspection with the repsective information that caused the flagging. As this reduced the amount of accounts that have to be checked manually, we were able to reduce the amount of manual work per sign-up substantially.

[1] http://www.izettle.com
[2] http://www.squareup.com
[3] http://www.sumup.com

Our second approach focused on reactive fraud detection and makes use of the most popular method to authorize credit card transactions in mobile payments: by signature. Authorization of credit card transactions can be done in various ways. Widely used are the three following methods:

- Swipe and signature (SAS): The cardholder swipes card through card reader and transaction is confirmed by signature on the mobile device. Low technical complexity but easy to attack as the data on the magnet stripe is not encrypted. Main problem: cards can be copied while the card data is read. Widely deployed in the USA due to high acceptance by Card Schemes.

- Chip and signature (CAS): The cardholder inserts card into card reader and the encrypted data is read from the card's chip. The user authorizes the transaction by signature on the smartphone screen. Higher technical complexity than SAS. Currently most popular method to authorize payments in Europe on mobile devices. Biggest drawback: Visa Europe doesn't consider this method secure enough and therefore requires an extended flow that requires the client to confirm the transaction with a text message.

- Chip and pin (CAP): User inserts card into card reader and enters pin into pinpad to authorize the transaction. Only after the correct pin has been entered, the phone is able to read the necessary data to process the transaction. Biggest drawback: High technical complexity as well in the reader as well as on the server side and thus high costs incurred by the hardware and software. Although this process is accepted by most card schemes in Europe, the high cost make a widespread use unlikely.

Although authorization with CAS is a lenghty process with VISA cards in Europe, it is currently the best compromise between speed, cost and usability. SumUp provides merchants with SAS readers or CAS readers based on the merchant's profile. This means that currently, all signatures are authorized by signature. Also with other providers of the same service, authorization by signature stays the predominant way to authorize transactions.

We propose an automatic signature verification system based on global featuers, dynamic time warping (DTW) and hidden markov models (HMM) to make the authorization process more secure. As the signatures which are captured by finger, tend to have a high variability and unstable characteristics, our system has to account for false positive fraud detection. To account for that, we propose a system involving a feedback loop. Using the feedback loop, transactions are still possible even if a cardholder's signature could not be matched to his previous signatures.

1.1 SumUp - Mobile Payments for Everyone

Credit cards enjoy a huge popularity among consumers which is illustrated by the number of credit card holders in the USA only: In 2008 over 176.8 million people owned a credit card with average of 3.5 cards per cardholder. About 60 percent of consumers own a rewards credit card and approximately 51 percent of the U.S. population has at least two credit cards.[1]

While credit cards are also very popular with consumers in Europe, far fewer businesses in Europe allow customers to pay by credit card than in the USA. This is mostly due to the following reasons:

- A european business initially pays between 200 to 500 Euro for a credit card terminal
- There is a monthly subscription fee from 20 to 50 Euros
- A percentage of each transaction goes to the credit card company. Usually, merchants pay between 2.75 and 5 percent to the card issuer.

It is likely that these costs are higher due to more complex regulations and more complex payment terminals that are required by the card schemes in Europe.

SumUp's goal is to lower the barrier for merchants to accept credit card payments by providing a professional yet inexpensive solution for anyone to accept card payments. The company was founded in fall 2011 and has enjoyed a rapid growth since. in Berlin and Dublin and is today used actively by many thousand merchants in more than ten european countries. Today, SumUp's services are used by thousands of merchants in more than ten european countries. The business has been successful because it removed two of the above-mentioned three obstacles to a more widespread adoption: anyone who signs up with SumUp receives a free card reader for use with one's smartphone and there is no monthly subscription fee. As the merchant already owns the expensive hardware in form of a smartphone and pays for a data subscription anyway, the cost on both ends is much lower. This way, SumUp is able to finance itself through the 2.75 percent transaction fee. At its core, it competes with traditional Credit Card Terminal companies who require their users to pay a monthly fee for their terminal and the expensive initial charge for the device.

Instead of building and selling expensive hardware, the only hardware required — the card reader — is shipped for free and the software is distributed for free via the Apple AppStore and Google Play Store. Payment through SumUp gives taxi drivers, market traderrs and other small stores, who couldn't afford one of the traditional payment terminals, an enormous economic advante, in that mobile payments can now be processed right on the spot and without an initial financial investment.

Processing a transaction with SumUp can be described with six simple steps as depicted in Fig 1.1 and ??.

1. Once the merchant has registered a SumUp Account and handed in identification documents, he receives a free card reader and can accept payments.
2. The purchase amount can be entered manually into the app or via predetermined product categories (Fig ??)
3. Debit cards and credit cards like Visa and MasterCard are all supported and can be chosen by clicking on the respective image. (Fig ??)
4. Once plugged in, the card reader confirms that it is ready to read a card. (Fig ??)
5. After reading the card, the app shows a confirmation of the card type and number. (Fig: ??)
6. The customer confirms the transaction with a signature written onto the screen of the smartphone or tablet. (Fig: ??)
7. After successful completion, the customer can have the receipt sent to their email account or via SMS to their phone. (Fig: ??)

On the server side, a variety of Fraud checks are performed and the transaction is either accepted or declined. The signature detection developed during this work will likely be implemented as an additional security measurement on the server side.

Besides it's simplicity and low setup time and cost, SumUp's advantage over its competitors is that it only collects a fixed percentage per transaction of 2.75%.

In the long term, SumUp will enable customers to process mobile payments, not just credit card transactions. As listed by the European Payments Council (EPC) [2], mobile payments may at one point replace all current Payment Methods for Business-to-Consumer (B2C) transactions as well as Business-to-Business (B2B), Consumer-to-Business (C2B) and Consumer-to-Consumer (C2C) transactions as shown with examples in Table 1.2. The EPC predicts this will happen because the availability and convenience of a mobile device paired with the perception of having full control over it, creates an environment of trust and convenience in which one conducts payments.

1.2 Fraud

Fraud, as defined by Phua et al. [3], refers to the abuse of a profit organisations's system without necessarily leading to direct legal consequences. Fraud Detection,

Figure 1.1: In-App Flow during a Transaction from step 2 (top left) to step 7 (lower right)

C2C	Repay a friend
C2B	Buy Groceries
B2C	Pay for train to work with company account
B2B	Pay for Business Lunch

Figure 1.2: Examples of mobile payment applications as listed by the EPC

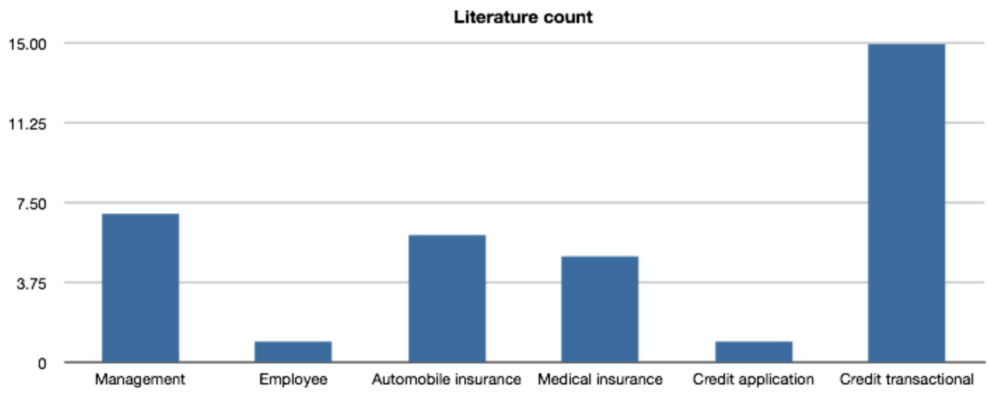

Figure 1.3: Number of research papers published on different fraud areas

as part of the overall fraud control, has become one of the most established data mining applications.

Automatic payment systems are always interesting for attackers due to their direct monetary return and usually high scalability. SumUp is no exception and is therefore interested in having a solid and extensive anti-fraud system.

As credit cards are historically a very unsafe payment method, special attention has to be paid when dealing with credit card information and processing credit card transactions. A fact that is reflected by looking at how many papers were written about fraud detection in which area. As shown in Figure 1.3.

The main attack vectors in credit card fraud are: (TODO: ref paper)

- Copied and stolen cards
- Money Laundry
- People transferring money under someone else's name
- Illegal money being transferrend through one's system

We will mainly focus on the first fraud case and concentrate on identifying a card holder by his signature. Our goal is that after a card has been used a certain amount of times, we can build a reliable signature model to identify whether it is the same person signing or not the next time the card is used.

Not only the pysical cards are at risk to be stolen, also the digital copy of the credit card data needs to be protected. This is one of the reasons SumUp only saves encrypted card information and does so in a PCI-DSS environment. The ensure maximum security precaution, only card numbers and not names of card holders are saved. This has one downside: We can only identify cards but not card holders and can therefore only build a signature for a card but not for a person (who might use several cards on several occasions).

Realisticly, it is impossible to be absolutely certain about the legitimacy of a transaction. In reality, the goal must be to filter out possible evidences of fraud from the available data using cost-effective mathematical algorithms.

PCI-DSS

1.3 Signature Recognition

Methods for signature detection is usually divded into Online and Offline Methods.

Offline Signature Detection performs recognition algorithms based on static features of a signature, mainly it's shape and length.

Online signature detection has become possible when digital tablets and touchscreens have become widely used and it became feasible to also capture the dynamic features of a signature.

In chapter X (todo: ref) we look at the common techniques in both areas to compare signatures.

Traditionally, digital signatures were captured on a digital tablet with a pen.

Signatures captured on mobile devices are different from existing work because

- they're capture by finger instead of a pen
- the sampling rate isn't constant as the finger-up/-down are event based
- the signature is captured on screen with different resolutions and densities

We discuss our strategies to overcome these challenges in chapter X (TODO ref).

Chapter 2

Fraud Prevention PEP & HRA Detection

The preventive part of our Fraud Protection work happens before a client is onboarded and therefore before he even has access to the system.

Any financial institution has a self high interested, to implement as many anti-fraud mechanisms as posible. Still, as the financial industry is a strictly regulated industry, many of these precautions are mandatory and described by the governing financial institution.

The financial institution governing SumUp's operations is the English Financial Services Authority (FSA) and the regulatory environment is specified in the Money Laundry Regulations from 2007[4]. Chapter 2 specifies the due diligence that has to be done for every client. As part of the due diligence process, the financial institutions are required to check that the customer does not fall into one of the two following categories:

- **Politically Exposed Persons** (PEP) are people who hold a prominent political function. In the United Kingdom this is defined as a national position. Also a PEP's spouse and children are included in this category.
- **High Risk Accounts** (HRA) are people who have previously commited a financial crime, been involved in a money-laundring related crime (e.g. dealing with narcotics) or are listed on a terrorism list.

Any new client needs to checked against both lists at sign-up and regularily after the initial sign-up.

For an identified PEP, additional due diligence needs to be done to onboard the client. Additional documents need to be collected that give detail information about the wealth and assets of said person. After performing the additional due diligence, a client can be onboarded but settlements need to be blocked.

Field	Value
uid	unique identifier within the World-Check database
last_name	subject's primarily used last name
first_name	subject's primarily used firs first name
aliases	other first/last name combinations that the subject has used
alternative_spelling	alternative spelling of the subject's name
category	used to define if subject is a PEP or HRA
sub_category	used to define if subject is a PEP or HRA
age	age of the subject
dob	the subject's date of birth
deceased	information about the subject's death
locations	the cities the subject was associated with
countries	the countries the subject was associated with
further_information	description and references about the subject
external_sources	information about from which sources the information in the database was extracted

Table 2.1: Fields of the World-Check database CSV file

A positiv match with the HRA list means that the subject cannot be onboarded and a Suspicious Act Report (SAR) needs to be filed with the Serious Organized Crime Agency (SOCA).

A test based on first and last name was already in place but produced a lot of false negatives and part of our work was to replace the existing test component with a more efficient one.

In the next section we'll present the data of the two databases and their origin before talking about the way we structured the lookup and implementation of our solution.

2.1 PEP & HRA Database

World-Check [5] is one of the provider's of the PEP & HRA database, which we chose to work with to get the necessary data.

The data is supplied in csv format and contains in 20+ fields. The most important fields are listed in table 2.1.

The full database file is about 3 Gigabytes big and cointains some 1.8 million records.

2.2 Algorithm

The algorithm we used to make the lookup faster, consists of two parts - parsing and lookup.

Parsing the data

For faster lookup, each record from the original csv file is parsed into multiple records in the database such that there exists one database record for each pair of first / last name combination from the set of first and last names in aliases and alternative spellings.

Listing 2.1 shows the algorithm.

Listing 2.1: The Algorithm that parses the csv file creates a separate DB record for each first/last name pair

```
for record in csv-file

    # Create Arrays to store all first & last names
    # and initialize with the most common first/last name
    first_names = []
    first_names << record["first_name"]

    last_names = []
    last_names << record["last_name"]

    # Get all other pairs from alternative_spellings ...
    for pair in record["alternative_spellings"]
        first_names << pair[0]
        last_names << pair[1]
    end

    # ... and from aliases
    for pair in record["aliases"]
        first_names << pair[0]
        last_names << pair[1]
    end

    # Now create the DB records
    for fname in first_names
        for lname in last_names
            if record["category"] == "PEP"
                PEP.create_in_db fname, lname, record
            else
                HRA.create_in_db fname, lname, record
            end
        end
    end
end
```

The database has an index on first and last name to speed up the lookup.

Lookup

To reduce the number of false positives, the lookup uses additional information besides the first and last name. For a lookup, the following information needs to be provided:

- First and last name
- Date of birth
- City
- Country

The algorithm is outlined in Listing 2.2. It's output is a set of Know Your Customer (KYC) action and statuses which flag the customer in the system and may restrict him from certain actions, e.g. processing transactions with his account.

Listing 2.2: The Lookup algorithm

```
def lookup fname, lname, dob, city, country

    # collect all hits on first and last name
    all_hits = verified_hits = []
    all_hits << PEP.find_by_first_name_and_last_name(fname, lname)
    all_hits << HRA.find_by_first_name_and_last_name(fname, lname)

    for record in all_hits

        # skip if peson is already deceased
        next if record["deceased"]

        # parse dob. if the dob in the db is incomplete (eg
            1971/02/00)
        # only check the complete parts
        y = record["dob"].year  != 0 ? record["dob"]["year"]  :
            null
        m = record["dob"].month != 0 ? record["dob"]["month"] :
            null
        d = record["dob"].day   != 0 ? record["dob"]["day"]   :
            null

        # now skip if one of the values exists but doesn't match
        next if y && y != dob["year"]
        next if m && m != dob["month"]
        next if d && d != dob["day"]

        # at this point, dob matches or wasn't given
        # now, check city
        next unless record["locations"].contains? city
```

```
        # and country
        next unless record["countries"].contains? country

        # all checks passed, add this record to the result set

        verified_hits << record

    end

    # now set KYC status for the proven records
    for r in verified_hits
        set_kyc_status r
    end
end
```

2.3 Implementation & Architecture

The PEP & HAR check was implemented as a Sinatra Ruby Applicaton, that is accessible through a REST interface. It uses ActiveRecord (AR) to connect to the database and runs two cron jobs to perform these tasks:

- **daily:** download of incremental update files from world check and including the incremental data into the database
- **weekly:** re-check of all existing customers in the system against the database

TODO: Add System Topology

Chapter 3

Fraud Detection Signature Verification

In this chapter, we'll present an overview of existing signature detection methods and available resources.

Traditionally, detection methods can be assign to either feature- or function-based methods. We describe both approaches in (TODO ref). A combination of feature- and function-based approaches has been providing better results than the individual techniques. (TODO: Ferrez-Aguilar et al, 2005)

3.1 Previous and Related Work

During the past 3 decades, a lot of work has been done on offline and online signature detection algorithms and many techniques exist.

Among those techniques, there's been work on Dynamic Time Warping [6] [7], Hidden Markov Models [8], directional pdf [9], stroke extraction [10], synthetic discriminant functions [11], granulometric size distributions [12], neural classifiers [13], wavelets[14], grid features[15] and elastic matching[16] to name a few. [17]

We will concentrate on the first two in our experiments as they have proven to be most successful and also most popular in earlier work.

An overview over previous work was given in a paper by Guo et al. [18]

TODO: list a few papers and their methods. Some for online and some for offline verification. List also how well they were performing.

3.2 Features and Feature Extraction

As introduced in chapter X (TODO), a features can be classified as a local or global feature. Global features describe the signature as a whole. Examples are

Table 3.1: Important features

Max acceleration	Avg. acceleration
Var. of acceleration	Avg. x acceleration
Avg. y acceleration	Total dots recorded
Time of second pen down	Avg. velocity
Max velocity	Avg. velocityÃ·max velocity
Avg. x velocity	Avg. y velocity
xy velocity correlation	Max x velocity
Max y velocity	Var. of x velocity
Num. pts. with positive x velocity	Var. of velocity
RMS velocity	Direction at first pen down, first pen up
First moment	Signature height (H)
Signature width (W)	W to H ratio
Pen down samples (D)	Time of max velocityÃ·D
Time of max x velocityÃ·D Point max. velocityÃ·D	Avg. azimuth
Avg. elevation	Number of pen ups and downs
RMS acceleration	

the discrete Wavelet Transform (TODO: ref), thue Hough Transform (TODO: ref), horizontal and vertical projectsions (TODO: ref) and smoothness features (TODO: ref).

3.3 Feature-based Systems

Feature-based systems, also called global systems, are characterized by the fact that the feature vector consists of measurements that are based on the whole signature. Example features are listed in table (TODO: ref)

Sequential Forward Feature Selection (SFFS) is one of the best performing methods (TODO: Jain and Zongker, 1997) but many have been proposed. The matching is usually done using statistical classifiers such as Parzen Windows (Martinez-Diaz et al, 2007), majority voting (Lee et al, 1996) Mahalanobis distancis (Galbally et al, 20007) or Guassian Mixture Models (Martinez-Diaz et al, 2007).

Important features include: [19]

Table 3.2: Important functions

Signature length	Horizontal position
Vertical position	Normal pressure
Path tangent angle	Initial direction Total velocity
x velocity	y velocity
Total acceleration	Pen elevation
x acceleration	y acceleration
Log radius of curvature Pen azimuth	

3.4 Function-based Systems

Function-based systems, also called local systems, are characterized by the fact that the feature vecotr consists of measurements on partial segments of the signature. Theses segments can be single points of groups of points. The most popular methods are Dynamic Time Warping (DTW) and Hidden Markov Models (HMM).

Important functions include: [19]

3.4.1 Comparison of time series

The most simple approach to compare two time signals that comes to mind, might be to use linear correlation [20] ie by comparing two time series dot by dot and calculating the distance between each pair.

But as soon as the time signals are not of equal length or there is a non-linear distortion, this approach will not be valid anymore. As it is very likely that the same signer's signature will have different dynamics every time he signs, this is not a feasible approach.

We discuss approaches that take these limitations into account in the next two sub-chapters.

3.4.2 Dynamic Time Warping

Dynamic Time Warping is a dynamic programming algorithm to measure the similarity between two time series which may vary in time or speed. This has been used for speech recognition and can also be used for signature detection to cope with the non-linear time distortions which one might see in the signals because a signer does not always sign with the same speed. It has shown to be a much more robust distance measure than the Euclidean distance [21][22][23] due

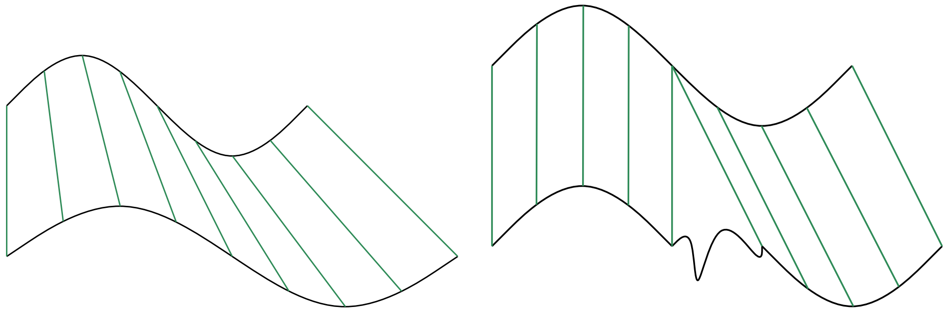

Figure 3.1: DTW allows to match two time series even if they are stretched (left) or distorted (right)

to it's ability to match similar shapes even if they are out of phase in the time axis.

Koegh et al. showed on a very large dataset that the mean error rate average over 1000 runs for DTW was an order of magnitude lower than the error rate for the Euclidian distance. However, the DTW algorithm also took approximately 230 times longer than the Euclidean distance. [24]

It has first been applied to signatures in 1977 by Yasuhara and Oka [25] who concluded that is a very useful approach for online real-time signature verification. Yasuhara and Oka used an adaption of the algorithm that was originally poposed by Sakoe and Chiba [26].

DTW allows us to compare signatures even if the signer was signing slower at the beginning in one of the two signatures.

Training is done by computing the distance measure $dtw[n][m]$ for all signatures n, m in the set of signatures for a certain user and selecting the signature s with the smallest distance to all other signatures.

Classification is done by computing the distance DTW distance $dtw[s][t]$ between the model signature s and a test signature t.

Algorithm We have two Signatures X, Y defined as follows:

$$X = x_1, x_2, ..., x_i, ..., x_I$$
$$Y = y_1, y_2, ..., y_j, ..., y_J$$

and a distance measure defined as

$$d(i, j) = ||x_i - y_i||$$

We define a warping path C as

$$C = c_1, c_2, ..., c_k, ..., c_K$$

where each element c_k correspondes to a combination (x_i, y_i).

The algorithm is initialized with

$$g_1 = g(1,1) = d(1,1) * w(1)$$

where g_k is the accumulated distance after k steps and $w(k)$ is a weighting function that has to be defined. In each iteration, g_k is computed as

$$g_k = g(i,j) = \min_{c_{k-1}}[g_{k-1} + d(c_k) * w(k)]$$

until both Signatures X, Y have been traversed.

The normalized distance of the two signatures is therefore:

$$D(X,Y) = \frac{g_K}{\sum_{k=1}^{K} w(k)}$$

where $\sum w(k)$ compensates the effect of the length of the sequences.

The definitionen of weighting factors w_k defines the matching between the two signatures. The most common definitionen in literature is such that only three types of transitions - deletion, instertion and match - are allowed. The resulting g_k becomes:

$$g_k = g(i,j) = \min \begin{bmatrix} g(i, j-1) + d(i,j) \\ g(i-1, j-1) + 2d(i,j) \\ g(i-1, j) + d(i,j) \end{bmatrix}$$

TODO: show image like 10.1007 s10115 page 362

Even though the DTW algorithm has been outperformed by more powerful algorithms like HMMs or SVMs in speech detection, it remains very effective in Signature detection as it deals well with small amounts of training data, which is typical for signature verification problems.

In general, DTW is known to have two drawbacks in signature verification:

- heavy computational load
- warping of forgeries

DTW causes heavy computational load becauseit does not obey the triangular inequality and thus indexing a set of signatures takes a lot of time. As soon as the pool of signatures for a signer get bigger, the computation costs raise because the test signature has to be compared to each of the signatures in the pool of confirmed signatures. Eamonn Keogh et al. presented a lower bounding method to index all samples without comparing each of them to each other. [24]

The second drawback can be addressed by looking at how straight or bended the warping path is. A straight warping path indicates that a genuine signature is more likely whereas a curvy warping path indicates a forgery. Work on this has been done [27] but made comparison between different signatures harder because it introduce another dimension and thus made computation harder and has hence not found wide spread use.

Hao Feng et al. proposed another extension of the DTW algorithm, called extreme points warping (EPW) which proved to be more adaptive in the field of signature verification than DTW and reduced the computation time by a factor of 11. [28] Instead of warping the signature as a whole, they only warp so called Extreme Points that are characteristic to a signer's signature and match the curves between those points linearly.

3.4.3 Hidden Markov models

A Hidden Markov Model (HMM) is a stochastic process with an underlying Markov Model of which the states can not directly be observed but the only observations can be made. Each state transition emits a certain observation with a certain probability.

Other than with a Markov Model, the states of a Hidden Markov Model can not directly be observed. Only Symbols, which are emitted from each state with a certain probability, can be observeed.

Markov Models can be modeled as Left-to-right, ... (TODO: name all types, show graph).

While HMMs with a too small set of states and observations perform bad because they are too simple, too many states and observations make the model computational heavy and accuracy is reduced because of overfitting.

There are three fundamental problems for HMMs: Given the model parameters and observed data, estimate the optimal sequence of hidden states. Given the model parameters and observed data, calculate the likelihood of the data. Given just the observed data, estimate the model parameters. The first and the second problem can be solved by the dynamic programming algorithms known as the Viterbi algorithm and the Forward-Backward algorithm, respectively. The last one can be solved by an iterative Expectation-Maximization (EM) algorithm, known as the Baum-Welch algorithm.

Baum-Welch-Algorithm

Algorithm: An HMM is defined by the following elements:

The HMM consists of N states $w_1, w_2, ..., w_N$, each of which cannot be directly observed. Each of the states emits a symbol at time t which can be observed. Therefore, after going through T time steps, the System emits the

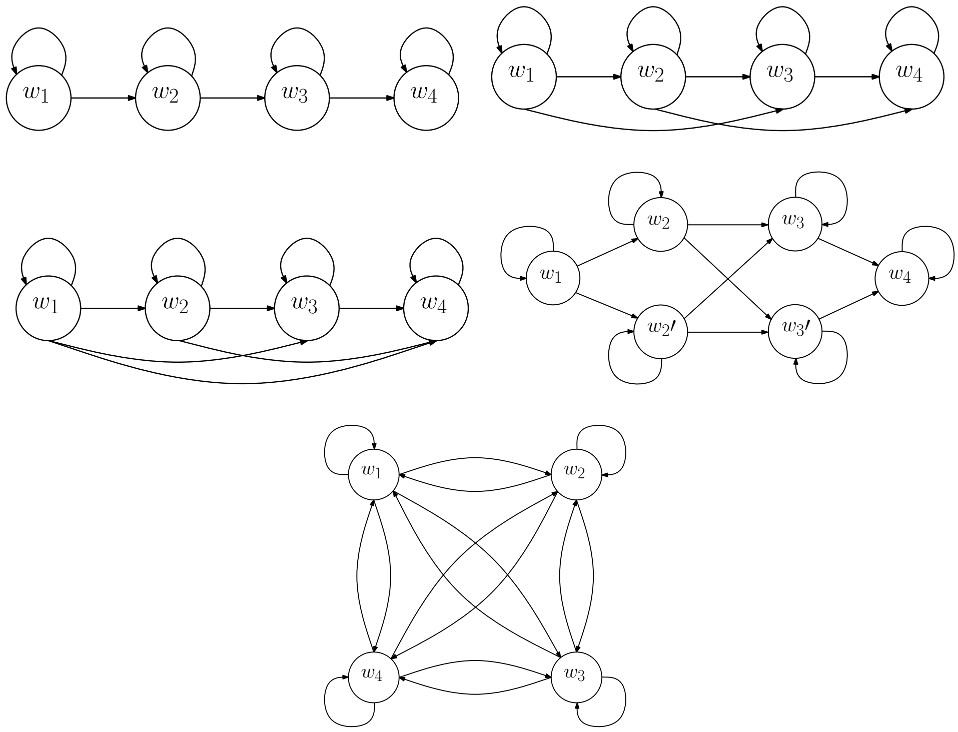

Figure 3.2: Different Markov Models: a: ...

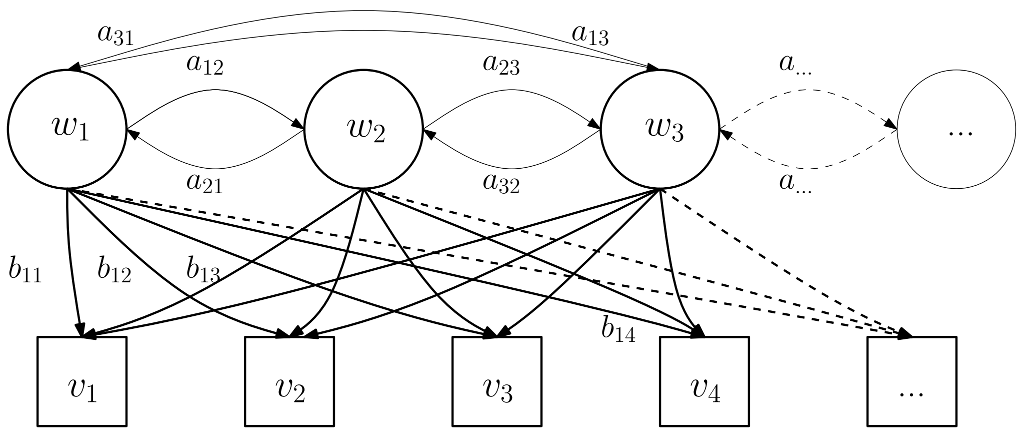

Figure 3.3: General representation of an HMM with states $w_1, w_2, 2_3, ...$ and observations $v_1, v_2, v_3, ...$

sequence

$$V^T = \{v(1), v(2), ..., v(T)\}$$

The probabilities to go from one state to another are given by the Transition matrix A with elements

$$a_{ij} = P(w_j(t+1)|w_i(t))$$

The probability that a system is a certain state when an certain symbol is observed, is defined by the emission matrix B with elements

$$b_j(v(t)) = P(v(t)|w_j(t))$$

A and B fulfill the normalisation condition:

$$\sum_k b_j(k) = 1, \sum_{j=1}^{N} a_{ij} = 1$$

π defines the initial probability distribution for a csystem to be in a certain state.

To summarize, an HMM is defined by the transition matrix A, the emission matrix B and the initial distribution π.

Therefore,

- Number of (hidden) States: N
- Number of Symbols per State: M

- Probability transition matrix $A = \{a_{ij}\}$ which defines the probability of going from one state to any other state or itself and holds $\sum_{j=1}^{N} a_{ij} = 1$ for all states i

Training:

3.5 Challenges in Signature Detection Specific to Mobile Payments

A lot of work has been done on signature detection on static signatures and also on dynamic signatures. However, almost all work on dynamic signatures has been done on signatures that were captured using a pen on a digital tablet. In our case, signatures were captured on a wide range of different devices and with a user's finger instead of a pen. This has several implications.

Our experiments in chapter X (TODO ref) will show to which extent known techniques are applicable to this type of signature and what future work might need to be done.

3.5.1 Signature are written with Finger

Our experience shows that people are not used to write their signature with their bare finger and the first few times they sign, their signature differs a lot. However, after just 10-15 times , the signature's shape seems to stabilize.

This means that it will be a lot harder to detect signatures based on the first few signatures than on later signatures, once a user got used to signing with his finger.

It also means that we should prefer later signatures to earlier ones as in later signatures the signer's signature might have stabilized.

3.5.2 Sparse Initial Dataset

Although SumUp is live in over 10 countries as of today, it is still only used in a relatively small set of locations and we are therefore unlikely to gather a lot of data about a certain user until the concept becomes more often deployed and used.

This means that we have to try and find a solution that works reasonably well with a small amount of initial data per user.

3.5.3 Device & Software Fragmentation

Unlike the signature's that are captured on a digital tablet, our database of signatures is captured on a variety of devices with different properties. There are various factors that have an influence on the digital representation of the signature:

- Different Manufacturer: Both, iOS and Android devices use a variety of manufacturers for their handsets and the touch screens used. Recent studies have shown the differences of how signals are captured on different screens (TODO: link reference)
- Screen Size: the screen diagonal of current smartphones typically ranges between X and X cm, those of tablets typically ranges between X and X cm. A consequence is that the user might not only sign slightly differently but also that the signature will consist of more or less data points and it will take users a longer or shorter amount of time to sign.
- ...

We will consider these factors when applying our algorithm in Chapter X (TODO)

3.6 Feedback Loop

Chapter 4

Experiments

Section 4.1 will give insight into how our work has affected the number of false positives and thus the manual inputs from Operations have changed since the new component was released.

Section 4.2 will first introduce the database of signatures we collected to test our signature verification algorithms and afterwards show the equal error rate (EER) achieved in different scenarios.

4.1 Preventive Fraud Detection

The algorithms described in Listing 2.1 and Listing 2.2 were implemented in a Ruby Sinatra app with a REST interface and released to production about 2 months prior to the submission of this thesis.

4.1.1 Setup

The full database was parsed on a testing environment and the resulting data was copied over into production. The daily updates are received on the production machine and directly parsed into the production database through the parsing script.

The PEP/HRA check is performed either automatically or manually:

- **Automatically** for each new user during sign-up
- **Automatically** for each existing user once a week to re-check each account after incorporating the changes of that week
- **Manually** from the Operations Admin Panel (OAP) via a button

Each request must be triggered via a POST request. Depending on the user object that needs to be checked, the body of the request changes. Table ?? lists the different request types.

Method	Path	Parameter
POST	/person/check	{"params": {"user_id": "1"}}
POST	/person/check	{"params": {"business_owner_id": "1"}}
POST	/person/check	{"params": {"business_officer_id": "1"}}

Table 4.1: The requests received by the PEP/HAR component

Device	Software Version	Screen Size [cm]	Screen Resolution [px]	Pixel Density [ppi]
Apple iPhone 4s	iOS 6.1.2	8.9	640x960	326
Apple iPhone 5	iOS 6.1.2	10	640x1136	326
Samsung Galaxy Note II	Android 4.1.1	14.1	720x1280	267
Apple iPad mini	iOS 6.1.2	20	768x1024	163

Table 4.2: The four devices used to collect signatures, ordered by screen size

4.1.2 Results

Show how many false positives/false negatives were collected before switch to new component and after switch.

4.2 Reactive Fraud Detection

4.2.1 Database

As there don't seem to be any publicly available databases of signatures that were collected with a finger on a mobile device, we collected our own database and forgeries.

Between 8 and 80 Signatures per person were collected from 11 people on 4 different days on four different devices. In total, a set of 487 signatures was collected.

The devices used to collect the signatures are listed in Table ??.

We also collected three forgeries for each person. The forgeries were created by an untrained person in the following modi.

- **Spontaneous Forgery**: A random signature from the set of one person's signatures was shown the imposter for 5 secs. The imposter was only allowed to forge the signature after the five seconds.
- **Sample Forgery**: A random signature from the set of one person's signatures was shown to the imposter for an indefinite amount of time and the imposter was able to look at the signature while forging the signature.
- **Knowledgable Forgery**: The imposter had unlimited access to all of one person's signatures to forge the signature.

Pre-Alignment and Normalization

4.2.2 Forgeries

4.2.3 Experiment Setup

Feature Extraction

Dynamic Time Warping

Hidden Markov Models

list different packages that are available and why we chose to work with matlab

Model

Training

Evaluation/Matching

4.3 Evaluation

4.3.1 Computational Requirements

4.3.2 Modi

4.4 Discussion and conclusions

4.4.1 Signature Detection

4.4.2 Feedback Loop

4.4.3 Future Work

- use lower bound proposed by Keogh et al.[24] to make DTW faster on large datasets

Chapter 5

Conclusions

Bibliography

[1] Woolsey, B., Schulz, M.: Credit card statistics, industry facts, debt statistics. Google Search Engine (2010)

[2] Council, E.P.: White paper mobile payments. EPC492-09 (October 2012)

[3] Wang, S.: A comprehensive survey of data mining-based accounting-fraud detection research. In: Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on. Volume 1. (2010) 50–53

[4] Authority, F.S.: The money laundring regulations 2007 (2013)

[5] world check.com: Aml/kyc/pep/cft solution — world-check marketing portal (2013)

[6] Herbst, B., Coetzer, H.: On an offline signature verification system. In: Proceedings of the 9th annual South African Workshop on Pattern Recognition. (1998) 600–604

[7] Yoshimural, M., Yoshimura, I. In: An application of the sequential dynamic programming matching method to off-line signature verification. Volume 1339 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (1997) 299–310

[8] Justino, E.J.R., Yacoubi, A.E., Bortolozzi, F., Sabourin, R.: An off-line signature verification system using hmm and graphometric features (2000)

[9] Drouhard, J., Sabourin, R., Godbout, M.: A neural network approach to off-line signature verification using directional PDF. Pattern Recognition **29**(3) (March 1996) 415–424

[10] Lau, K.K., Yuen, P., Tang, Y.Y.: Stroke extraction and stroke sequence estimation on signatures. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. Volume 3. (2002) 119–122 vol.3

[11] Wilkinson, T.S., Pender, D.A., Goodman, J.W.: Use of synthetic discriminant functions for handwritten-signature verification. Appl. Opt. **30**(23) (Aug 1991) 3345–3353

[12] Sabourin, R., Genest, G., Preteux, F.: Off-line signature verification by local granulometric size distributions. Pattern Analysis and Machine Intelligence, IEEE Transactions on **19**(9) (1997) 976–988

[13] Bajaj, R., Chaudhury, S.: Signature verification using multiple neural classifiers. Pattern Recognition **30**(1) (1997) 1 – 7

[14] Ramesh, V., Murty, M.N.: Off-line signature verification using genetically optimized weighted features. Pattern Recognition **32**(2) (1999) 217 – 233

[15] Qi, Y., Hunt, B.R.: Signature verification using global and grid features. Pattern Recognition **27**(12) (1994) 1621 – 1629

[16] de Bruyne, P., Forre, R.: Signature verification with elastic image matching. In: Proceedings - 1986 International Carnahan Conference on Security Technology: Electronic Crime Countermeasures., Chalmers Univ of Technology, Gothenburg, Swed; Univ of Kentucky, Coll of Engineering, Office of Engineering Continuing Education, Lexington, KY, USA; IEEE, New York, NY, USA; GTE Service Corp, Evaluation & Support Dep (August 1986) 113–118

[17] Piyush Shanker, A., Rajagopalan, A.N.: Off-line signature verification using dtw. Pattern Recogn. Lett. **28**(12) (September 2007) 1407–1414

[18] Justino, E.J., Bortolozzi, F., Sabourin, R.: A comparison of svm and hmm classifiers in the off-line signature verification. Pattern Recognition Letters **26**(9) (2005) 1377 – 1385

[19] GÃ$\frac{1}{4}$ler, I., Meghdadi, M.: A different approach to off-line handwritten signature verification using the optimal dynamic time warping algorithm. Digital Signal Processing **18**(6) (2008) 940 – 950

[20] Plamondon, R., Lorette, G.: Automatic signature verification and writer identification - the state of the art. Pattern Recognition **22**(2) (1989) 107 – 131

[21] Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '00, New York, NY, USA, ACM (2000) 285–289

[22] Fuentes, M., Garcia-Salicetti, S., Dorizzi, B.: On line signature verification: Fusion of a hidden markov model and a neural network via a support vector machine. In: Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on. (2002) 253–258

[23] Shafiei, M., Rabiee, H.: A new online signature verification algorithm using variable length segmentation and hidden markov models. In: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on. (2003) 443–446 vol.1

[24] Keogh, E.: Exact indexing of dynamic time warping. In: Proceedings of the 28th international conference on Very Large Data Bases. VLDB '02, VLDB Endowment (2002) 406–417

[25] Yasuhara, M., Oka, M.: Signature verification experiment based on nonlinear time alignment: a feasibility study. IEEE Trans. on Systems, Man and Cybernetics, part C **12**(3) (1977) 212–216

[26] Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on **26**(1) (1978) 43–49

[27] Sato, Y., Kogure, K.: Online signature verification based on shape motion and writing pressure. Proceedings of the 6th ICPR (1982) 823–826

[28] Feng, H., Wah, C.C.: Online signature verification using a new extreme points warping technique. Pattern Recogn. Lett. **24**(16) (December 2003) 2943–2951

APPENDIX A

Appendix Chapter