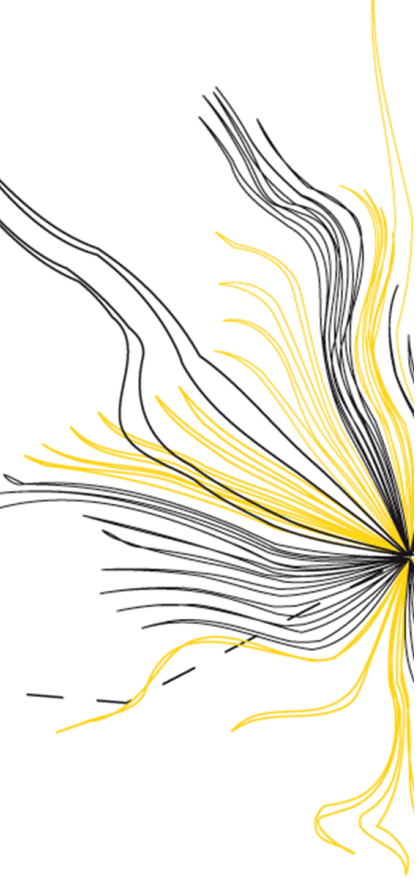


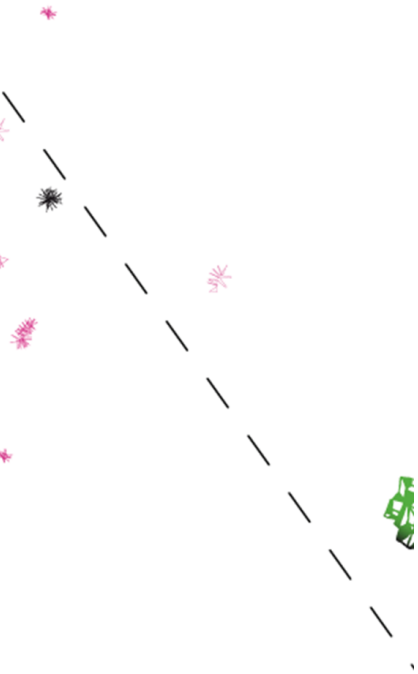
UNIVERSITY OF TWENTE.

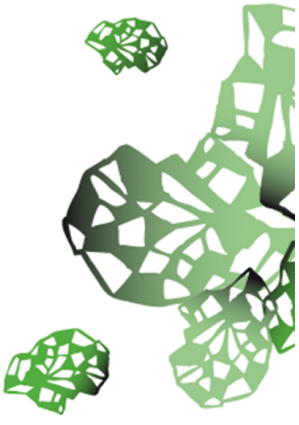
Faculty of Electrical Engineering, Mathematics & Computer Science

Anomaly detection in defence and surveillance

Steven Dirk Auke Sybenga

M.Sc. Thesis
August, 2016





ALTEN

TNO innovation for life

Supervisors:
Dr. M. Poel
Dr. G. Englebienne
Ir. R.L.F van Paasen
Dr. S.K. Smit

Human Media Interaction
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
7500 AE Enschede
The Netherlands

Preface

Books concerning state surveillance and their related ethical questions such as Cory doctorow's 'Little brother' and 'Homeland' as well as television series like 'Person of Interest' show a world where anomaly detection is used to catch criminals. Even though these stories are still science-fiction, reality seems to catch up while keeping it more or less secret for mankind. I am not a conspiracy thinker but mass surveillance is a reason for me to question whether some organizations and governments have or have not crossed an ethical border by their way of collecting data, what their motivation is to do so and how much they value the privacy of people.

Both the curiosity in what is actually possible of those scenarios and the challenges associated with it motivated me to do this final project when it was proposed to me by TNO. I have never changed my opinion about the risk of automation in surveillance. Every person is different and might for that reason be flagged as a person of interest. This does not imply any bad intention and therefore I would suggest to always have a human in the cycle to assess the actual risk this person is.

Acknowledgement

Productivity is often an issue for students writing their thesis, but thanks to the support from Alten in Apeldoorn that was not a problem for me. The ability to have a desk to work full time, the input and ideas from the people at the office as well as the coffee helped me a lot. Alten was also able to get me in touch with the right people at TNO which eventually led to this interesting project. Therefore I would like to thank Alten and TNO for giving me the opportunity to do this project.

Furthermore I would like to thank my supervisors for extensively correcting my writing, asking difficult questions and other support. Not only during this research

but during my whole study.

Last but not least I thank my family, friends and anybody who had either technical, ethical, financial, philosophical input or just curiosity in this project.

Summary

Security and crime prevention have always been a hot topic but with the recent rise of the number of terrorist attacks and the subsequent fear among people made it an important subject for police and military forces. Technological improvements of cameras and sensor technologies prove to be helpful in minimizing risks of attacks. Although terrors will always try to find evade devices such as metal detectors, x-rays and cameras, mentioned technologies have the potential to reduce the number of incidents.

Solving smaller crimes such as robberies and thefts are unfortunately daily business for police forces as well. The high number of incidents leave lots of victims traumatized or even wounded. Nowadays, highly populated areas will be covered by camera surveillance but in other areas, getting away with such an offend is still far too easy.

For both high and low impact crimes, technologies capable of detecting suspicious behaviour could reduce the occurrences or provide fast response in case of an incident. This research focusses on finding suspicious behaviour (anomalies) during tracking people in areas which have the size of multiple city blocks to complete city sizes, for example by detecting people using drone images or other tracking systems.

The goal of this research is to develop and evaluate an anomaly detection techniques capable to find abnormal or suspicious behaviour based on positions of people. Similar goals have been the subject of other researches, often resulting in one method of detection one type of anomaly. Although many methods are capable of performing anomaly detection, most times trajectory analysis in combination with statistical models show good results.

Part of the research is the development of anomaly detectors, capable of detecting anomalies in simulated data. To evaluate the detection methods, four cases

will be simulated: one in with no special event occurs, one with four processions, one with four street robberies and one in with four commercial stores or banks are robbed. Each simulation will take four simulated hours, spanning from noon till 16:00 with events occurring at 37 minutes after every hour. The events are planned on the hour to give the people time to get to the event.

The combination of detectors and methods designed were able to detect the offenders in all three crime related events. As for most related research, Gaussian based models were performing best when (abnormal) speed is used as feature. The context such as the location where a person is detected or the history of the person is an important factor in anomaly detection, especially when we are coping with big areas.

The detectors were tested with simulated data, this makes the results questionable for real life situation. In the simulator people always had a goal they walked to and did this in walking pace. Nobody was running to catch a train or as sporting activity, which makes detection of running offenders fairly easy. For this reason, detectors using different contexts were designed and evaluated as well.

Global detection works best for the simulated data due to the relatively constant non-anomalous behaviour generated by the simulator used for this research. Models based on personal trajectories works well for outdoor events but not for indoor incidents due to the lack of pre-event trajectories of the offender. Location based detectors are less successful compared to global detection for both situations due to the limited amount of training data.

Another type of detection capable of finding collective anomalies is able to reliable detect processions, based on the (change of) density at different locations. This method could also use context such as the time of day to detect whether a location is currently more crowded compared to the same time on another day.

Adding trajectory detectors based on other context such as time-of-day and collective detection based on (dis)similarity of people are recommended as future work for the designed system. Other recommendations include the evaluation of the methods on real life data, possibly with actors playing out the events and research on whether or not detected anomalies is suspicious behaviour according to a domain expert.

Contents

Appendices

List of Abbreviations

AARP	Automated Anomaly Detection Processor.
AD	Anomaly Detection.
AIS	Automatic Identification System.
ANN	Artificial Neural Network.
ART	Adaptive Resonance Theory.
DSDR	Discrete Spacial Distribution Representation.
FIFO	first in first out.
FN	False Negative.
FP	False Positive.
GMM	Gaussian Mixture Model.
GMTI	Ground Moving Target indicator.
HTM	Hierarchical Temporal Memory.
IDSA	Intent Driven Scenario Authoring.
iForest	Isolation Forest.
IQR	Inner Quartile Range.
ISR	intelligence, surveillance and reconnaissance.
K-NN	K-Nearest Neighbors.
KDE	Kernel Density Estimation.
KDF	Kernel Density Function.
LOF	Local Outlier Factor.

MLE Maximum Likelihood Estimation.

OD Outlier Detection.

PCA Principal Component Analysis.

PDF Probability Density Function.

PGA Peer Group Analysis.

SA Situational Awareness.

SNG Stochastic Neighbor Graph.

SOM Self Organizing Map.

SOS Stochastic Outlier Selection.

SVDD Support Vector Data Descriptor.

TP True Positive.

t-SNE t-Distributed Stochastic Neighbor Embedding.

TD Target Detection.

UAV Unmanned Aerial Vehicle.

VQ Vector Quantization.

Chapter 1

Introduction

Due to recent terrorist attacks, all security agencies are on high alert to find suspicious activities. Unfortunately no system will be able to prevent all possible threads. In the meantime, police also has to deal with smaller crimes which can still have significant impact on the victims. Technological innovations can be used to minimize both high impact terrorist attack and smaller crimes.

Increased quality of cameras (for example the ARGUS-IS [1]) enables the possibilities for intelligence, surveillance and reconnaissance (ISR) of wide areas. However, any operator looking at those live feeds would have no clue where to look at. Advanced image processing techniques can be used to detect objects such as people, vehicles etc. which increases the Situational Awareness (SA) of the operator. For wide area surveillance where the covered area is tens of square kilometres, it could potentially detect hundreds of objects at the same time. Since this is still too much information for an operator to cope with, an automatic preselection of persons of interest is required. Detecting abnormal behaviour (anomalies) in the tracked data enables the possibility to inform the operator where to focus on.

To understand what behaviour has to be considered anomalous, this paper will first explain the characteristics of different types of crimes and how it could be detected. Relevant research and literature will be consulted to explore the possibilities of using the behavioural aspect of the characteristics in anomaly detection. Furthermore the methods found will be part of a detector build to find anomalies in human behaviour. This analysis is one of the methods to assist in prevention or to provide quick response to such crimes. Other methods such as eavesdropping on communication channels enables police to understand and detect anomalous behaviour but is not part of this research.

1.1 Research questions

The goal of this research is to create and evaluate anomaly detectors usable for military and surveillance SA. The detector should be able to provide extra information on where the operator should focus attention to. Designing the visualization tool itself is not part of the research but for demonstration purposes, a representation of the anomalies and normal data will have to be presented.

The main research question will be as follows:

> *What techniques can detect anomalous behaviour of people based on the position and trajectories in an area of multiple squared kilometres.*

Calling an outlier generated by a detector an anomaly, is up to a domain expert. For this an operator will have to understand what kind of events are detectable by what anomaly detector. In other words, what does an outlier actually tell us in terms of human behaviour:

> *How do the outliers generated by the models of a detector relate to anomalies in human behaviour?*

Due to security and privacy issues, the research has to be evaluated using simulated data. Furthermore, a simulator can provide us anomalous events to evaluate the detectors where real data with those anomalies is hard to find. We will use a simulator provided to us by TNO, capable of generating such anomalous events. This does mean we have to have a closer look at the events generated by the simulator:

> *What anomalous events can be generated with the simulator and what do the events do?*

A system has to be designed to collect the data generated by the simulator and perform the anomaly detection. Since the ability to detect anomalous events in real time is crucial for operators, this will have to be a requirement for the system.

> *How do we design an anomaly detector capable of detecting the events in real time (online)?*

To evaluate the models, the designed detectors are fed with the simulated data. The results will answer the last sub-question:

> *How well are the previously mentioned methods able to find the generated anomalous events?*

1.2 Report organization

The remainder of this report is organized as follows. Chapter 2 describes different types of crimes and their corresponding human behaviour. In Chapter 3, relevant research is reviewed to determine what methods and techniques could be used to find anomalies and to answer the research question. Chapter 4 explains what methodology and principles were used during this research. In Chapter 5 the design of an anomaly detection (testing) framework is explained. This is used to test the detection methods on simulated data and the results of these tests are given in Chapter 6. Finally, Chapter 8 contains the conclusions and recommendations.

Chapter 2

Crimes

There is usually no exactly definable characteristic for the several criminal events, nevertheless it is possible to generalize certain characteristics to explain different types of crimes[1]. This chapter will cover two types of crimes: high impact crimes, affecting a high number of people such as terrorist attacks and low impact crimes like pickpocketing.

2.1 High impact crimes

A drastic increase of the number of fatalities caused by terrorist attacks supports the growing fear among people. Nine times more people died from terrorist related incidents in 2014 compared to 2000. There is an increase of 80% from 2013 to 2014, and due to recent attacks it is unlikely this trend will be broken soon. Close to 80% of the incidents occur in Syria, Iraq, Afghanistan, Pakistan and Nigeria but there is an increase in other countries affected by terrorist attacks [2].

Motivations to join a terrorist organisation or perform an attack differ for the type of organization (e.g. political, religious or ideological). However, there is a strong correlation between the country where the attack takes place and ongoing conflicts in or related to that country. Political instability and -terror as well as human rights issues and suppression of religious freedoms also correlates to terrorist attacks [2].

[1] Statistics about the occurrences of the events in the Netherlands can be found in Appendix C.

2.2 Low impact crimes

Although preventing terrorist attacks is a high priority of police forces, especially in the western world more people are victimized due to low impact crimes. The amount of occurrences is higher and therefore the number of victims too.

2.2.1 Procession and public gatherings

Processions are characterized by their collective behaviour of a big group walking slow. Processions are in most cases and countries not considered a crime but could potentially turn violent when riots start to form. The same holds for public gatherings in general, especially when no announcement of the event has been made. Research in crowd dynamics can prevent dangerous situations when large groups of people gather [3] but to accomplish this, prior knowledge or early detection of the forming of a crowd is needed.

2.2.2 Pickpocketing, robbery and street theft

A street theft can be done stealthy or violently depending on the type of theft. Pickpocketing is usually done without alerting the victim either by using stealth or distracting (a con). The offender can work alone or use a team in which one steals the valuable object while the second person walks away with it. On the other side of the spectrum are robberies as violent crime, which will leave the victim traumatized or possibly wounded by confrontation or blitz attack methods. Snatch-thefts are less violent quick methods where an item is taken from the victim without the use of verbal communication.

Half of the victims are physically attacked during a robbery and 20% left wounded. Robberies mostly occur during the evening and night, when young adults are a good target due to alcohol consumption and distraction. In the morning elderly people are often targeted and children are among the victims in the afternoon, when they go home after school. The locations most robberies occur are in urban environments, close to the victims homes. Parking lots, garages, parks, fields, playgrounds and near public transportation are other locations where robberies often take place. Street thefts are most common in medium density areas. In crowded areas the of-

fenders do have enough potential victims but they are also protecting each other. Low density areas are also uncommon because there are less victims, so offenders will not look for them here [4].

2.2.3 Shoplifting and commerical robbery

Shoplifting is the act of stealing products without paying for it. Since the actual crime is committed inside the shop, this will not be detectable. However, when the thief is caught steeling, expected detectable behaviour is having them run out of the shop. The same behaviour is expected when a robbery of a bank, gas station or convenience store takes place.

Shoplifting incidents occur often in the second half of the week and more when the demands of goods are high, such as during pre-Easter, -summer and -Christmas periods. Since shoplifters are often juveniles, non-school days and -times and locations close to schools have high amounts of shoplifting [5].

In the US, about 9% of commercial robberies were bank robberies. This percentage is higher in smaller cities (12%) compared to larger cities (8%) but larger cities do have significantly more bank robberies compared to smaller cities [6]. Most bank robberies are quick and without violence due to the compliant employees (as they are trained to do) and initially successful and lucrative as well. However, one third of the bank robberies are solved within a day and 60% will eventually be solved. Bank robbers often repeat successful methods, which can also help to solve previous robberies when offenders are caught.

Unlike to what is usually shown in films, most robbers do not use any disguise (60%), are unarmed (72%) and are alone(80%). These non-violent amateurs tend to commit their crime during busy hours where professionals are more likely to pick quiet times such as opening and closing hours. Solitary robbers will not use a getaway vehicle but escapes on foot (58%) where teams often use a car (72%). The necessity of running is minimized by picking a target with easy access to busy pedestrian traffic. Bank robberies have a high risk of repeated victimization, where successful robbers go back to the same location to rob it again or because of the vulnerability properties of the bank (easy access and escape routes, security and prevention methods etc.) [6].

2.2.4 Home and Vehicle Burglaries

Theft from cars is among the most often reported larcenies. Most thefts from cars occur in the late night, early morning. Thieves, often juveniles or drug addicts, will mostly steal car parts (stereo, airbag) or valuable personal items (wallet, phone, laptop, etc.) to sell them and facilitate their addiction [7].

Among burglaries in houses, a single family house is often an attractive target compared to other types such as apartments, flats and semi-detached houses. This is caused by the multiple entrances single family houses usually have, the lack or minimized risk of witnesses due to the distance to the neighbours.

Houses on the outskirts of neighbourhood (where a burglar does not stand out) are more likely to get burgled. For the same reason houses near busy streets have a higher risk. Poor lightning, concealed entry points and cover are important factors, especially because burglars commonly take the side or back door to get in. Familiarity for the offender is an key aspect in their choice in deciding what house they will go to. This can be a familiar house because it belongs to a friend or acquaintance or because it was burgled before. Repeated victimization is not only caused by familiarity, also the presence of new valuable items (replaced since the last burglary) and the easy access are reasons for an offender to return.

Burglaries often take place during the day, when the occupants are at work or during the night when they are sleeping. Burglars will look for several clues to see when the house is empty, such as accumulating mail, the lack of a car on the driveway, and no lights or sounds coming from the house. Routine in these clues will suggest the owners are at work or on holiday [8].

2.2.5 Bicycle theft and grand theft auto

Despite bicycle theft being accounted for a high number of the larceny incidents (4% in the US up to 25% in the Netherlands), few people report it to police. A Reason not to file it is the lack of trust in the police to solve the crime, catch the thief and return the bicycle. The main motivation for offenders to steal a bicycle is to get to somewhere quickly (joyride) or to sell it for money. The first one mentioned often refers to young offenders, on the other hand poor people and drug addicts steal to trade the bike for cash [9].

Cars are often stolen from the victim's home (37%) and more likely from the street than from a driveway or garage and more often at night when they are parked at those homes, as well as the cover for the thief due to the darkness. The neighbourhoods with a high number of potential offenders (usually the poor neighbourhoods) have a higher risk since thieves prefer to find a target close to their home. They know this area and do not have to walk far to find the car. Older cars are more prone to get stolen compared to newer cars, not only because they are more common in poorer areas but there is also lack anti-theft security in the cars to prevent it. Stolen cars are used for joyriding, other crimes (for example as getaway car), for reselling or to strip car parts [7].

2.2.6 Stalking

Stalking is an ongoing event and not a single identifiable crime like the offences mentioned before. No profile can easily be defined as there are lots of reasons why and methods how people stalk their victim. Stalking behaviour can be complex and can range from sending messages to following or assaulting the victim [10].

Chapter 3

Related work

3.1 Characteristics of Anomaly Detection

Research in Anomaly Detection (AD) focuses mostly on computer network intrusion but there are several other domains where AD is used [11], [12]. These domains all have their unique approach but the techniques used have common grounds in all domains.

The term anomaly and outlier are often interchangeable and will be used as such in this work. However technically speaking there is a difference between the two [13]:

- An anomaly is an observation or event that deviates quantitatively from what is considered to be normal, according to a domain expert.
- An outlier is a data point that deviates quantitatively from the majority of the data points, according to an Outlier Detection (OD) algorithm.

Therefore, the presented anomaly detectors are in fact outlier detectors until a domain expert agrees with it being anomalous. Any detected outlier which is not an anomaly is considered a false positive of the detector.

An AD problem can be specified by different factors: The input data, the anomaly type, the labels and the output [11]. Based on these factors we can compare what technique and approach is suitable to detect what type of anomaly (see figure 3.1).

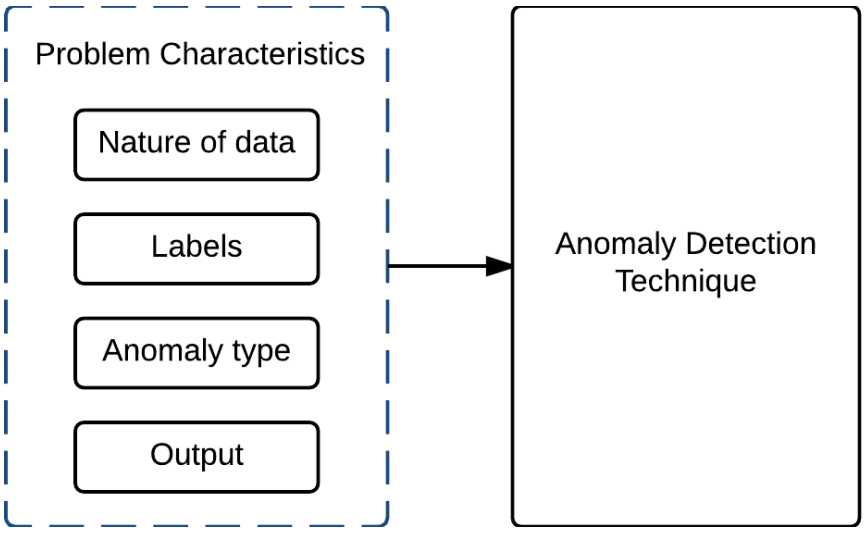


Figure 3.1: A close look at the characteristics of the problem is a good way to find the best AD technique for the problem.

When we are trying to evaluate multiple anomalies on the same dataset, the problem characteristics have to be assessed for every type of anomaly (figure 3.2). The labels as well as the output might be equal for some combination of anomalies, especially when no labels are predefined.

The combination of multiple AD techniques is used to give a single answer on whether an instance is an anomaly or not as can be seen in figure 3.3. This ensemble can have priorities or weights on what detector is more important because of the anomaly type it covers. Regardless of the way these weights are defined, adjusting them according to the preferences of the operator could be preferable. Feedback on which detector marked which instance as anomaly, and subsequently what anomaly type was detected is important information for the operator [14].

3.1.1 Input data

The input data for an anomaly detector for suspicious human behaviour are data instances representing a person or vehicle. These instances can be the result of a Target Detection (TD) algorithm, a sensor network, manual input, etc. An instance itself has different features, which are for example the position, speed, history (or path) and type (what kind of vehicle). All features might individually, or as a combination, be the input data of an outlier detection algorithm and are usually either binary, categorical or of continuous types [11].

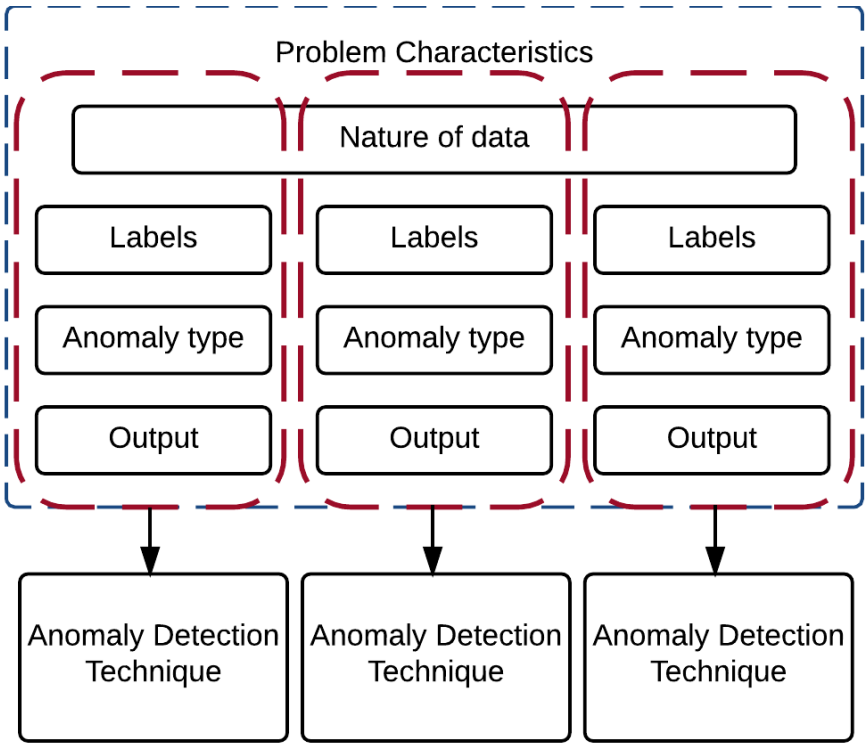


Figure 3.2: Characteristics of multiple ADs

Finding the right features is one of the most important aspects of anomaly detection [15]. Finding a good representation of the data into features is often challenging and can be the difference between a good detector and a useless one.

3.1.2 Labels

Another challenging aspect in AD is the lack of available labels of whether an instance is either normal or anomalous. The datasets, in which anomalies are by definition scarce, are usually big, therefore labelling all data as normal except those instances that are considered anomalous by a domain expert could be a solution. However, usually not all possible types of anomalies might exist in a dataset, can be predicted or defined. For those cases it is not possible to use supervised learning methods that would classify data by comparing the two different groups.

With online AD it could be possible to identify the false positives as they occur. Although much harder, even some false negatives are detectable when an operator notices anomalies in the big set of non-anomalous people. By labelling these incorrectly classified instance, a small shift from unsupervised to semi-supervised classification can be made or parameters could be tweaked by the system dynami-

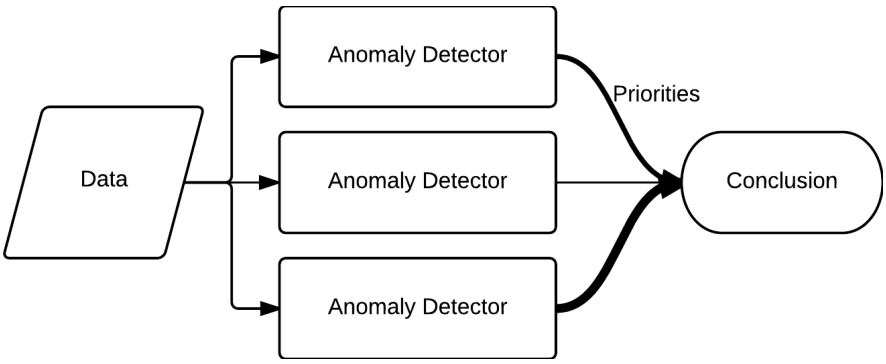


Figure 3.3: Combination of three anomaly detectors with unique weights (shown as arrows with different thickness).

cally, based on the label given by a domain expert [16].

3.1.3 Anomaly types

Anomalies can be grouped into four different types: point, contextual, spacial-temporal and collective anomalies. They are all detectable based on different assumptions.

Point anomalies

Point anomalies are instances that are outliers based on comparing their feature values to those of the complete dataset. They are, for example, extreme values that should not occur in any normal circumstance and are considered the simplest type of anomaly [11].

Contextual Anomalies

A contextual anomaly might look normal when compared to the whole dataset but is an outlier based on its context. Any knowledge about the data is required to define when instances share the same context. This can be a spacial distance between objects (neighbourhood), type or size of the object, etc.

An example of the complexity of contextual anomalies can be found in [17, p. 867]: Running could be defined as anomaly since most people walk instead of run in a normal situation. On a football field however, you will see the players often

running. This makes the pitch as context different for the behaviour running. Now imagine there is another event in the same stadium, for example a concert. In this context, running is suddenly abnormal behaviour again.

A common approach to eliminate this problem is to only test instances within the same context and test for semantic or class outliers [18]–[20]. Depending on what features the instances have as well as the detection technique, preprocessing might be required. However, several techniques can cope with contextual (or correlating) features and will not need preprocessing for context reduction.

Trajectories and spacial-temporal Anomalies

The trajectory (or path) an instance took to get to its destination can be used to detect anomalies as well. Instead of taking the whole path or history in consideration, it is also possible to look at associations between moments in time. For example, what is the likelihood of an instance going to location B if it passed A.

Collective Anomalies

The examples mentioned so far are anomalies detectable when looking at individual instances, where the instance itself is an anomaly. For collective anomalies it is not a single instance but a group that triggers a detector such as a crowd.

3.1.4 Output

The result of an anomaly detector could be a boolean value, specifying if some instance is an anomaly or can be a probability or 'score' of this instance being an anomaly. Using the latter as indicator for the operator has both advantages and disadvantages since a high probability does not imply a high priority but changing the score threshold can increase or decrease the number of (false) anomalies the operator sees.

Some anomalies such as running to catch the bus are not directly concerning but require attention nevertheless. Proper feedback on why something is marked as anomaly should be part of the system to decide if the anomalous person might be

up to no good. In other words, it is important for an operator to understand on what grounds an instance is marked as anomaly.

Negative consequences

AD has cases which can cause severe negative consequences when inappropriate decisions are made based on the anomalies detected.

A false positive can take the attention away from a serious anomaly. As long as the operator is able to quickly identify the alarm as false and able to focus on other detected anomalies, the consequence is a short delay in appropriate action. This is in most cases still better than evaluating every detected object but detrimental nevertheless.

3.1.5 Conclusion

One single outlier detection algorithm will most likely not be able to detect all different ways an instance can be considered anomalous. A committee of detectors, each designed for one or multiple anomaly types is required and also provides the ability to give feedback on what anomaly type is detected for the instance.

Detection of outliers is based on extracted features of the input data. Subsequently many anomaly detection methods can be used to determine what instances are outliers. The next two sections will respectively cover different possible feature extraction and anomaly detection methods.

3.2 Feature extraction and Preprocessing

Features for anomaly detection can come from sensor data or can be properties of the objects. Any type of feature value can be transformed into the other types. For example, a discrete value can be transformed into binary by using a threshold or into categories by averaging or by Vector Quantization (VQ). A colour, which is a categorical feature, can become a continuous value when the hexadecimal representation of the colour is used.

Extra preprocessing steps could transform features into different distributions. For example feature x_1 can be transposed into x_2 where $x_2 = x_1^y$ or $x_2 = \log(x_1 + y)$ (for a log-normal distribution) with any chosen value of y to create a normal distribution of the data.

3.2.1 Vector Quantization

Vector quantization is a optimization method in which the data is grouped based on their closely related or almost equal features. All points within a group can be represented as the centre (or prototype) vector of this group, compressing the size of the dataset [21], [22]. A simple example of Vector Quantization (VQ) is rounding of rational numbers to integers, where values 1.9 and 2.1 are grouped together having centroid value 2. VQ van be used as preprocessing or classification, such as the box plot explained in 3.3.1.

3.2.2 Sparse coding

The opposite of compressing feature values is done in sparse coding. The idea is to generate a sparse representation of the input which can reconstruct the original data. We assume the dataset to have a set of common descriptors, of which some combination of them generate the input. For example, images can be reconstructed by a combination of lines [23]. The vector representation of the weights will contain mostly zeros, and a small amount of non-zero elements for which the descriptor actually generates the input. If the set of descriptors (usually called the dictionary) is D and the sparse vector of weights corresponding to x^i is a^i, the input is reconstructed by $x^i = a^i \cdot D$.

If the sparse vector contains binary values, it can be presented by the indexes of the active bits in the vector which on its turn is a compressed representation of the dataset.

As for VQ, sparse coding is both a preprocessing method as well as a common step within ADs. For example Artificial Neural Network (ANN) (see section 3.3.4) and Hierarchical Temporal Memory (HTM) (section 3.3.4) make use of sparse representations.

3.2.3 Dimension reduction

Other preprocessing could be done by performing dimension reduction methods [24], where minimal data loss should be acquired [25].

Combining features

Some combination of features can have a strong correlation, for example due to their contextual property. In this case, a new feature based on the combination of the original features (for example feature $x_3 = \frac{x_1}{x_2}$) can be constructed and the distribution parameters for this new feature are to be found.

Principal Component Analysis

Principal Component Analysis (PCA), first mentioned in [26], is a method to find the linear component or hyperplane on which the dataset fits best. The first component represents the single direction with the most variance and the second represents the direction of the most variance relative and orthogonal to the first component (as can be seen in figure 3.4) [23], [24].

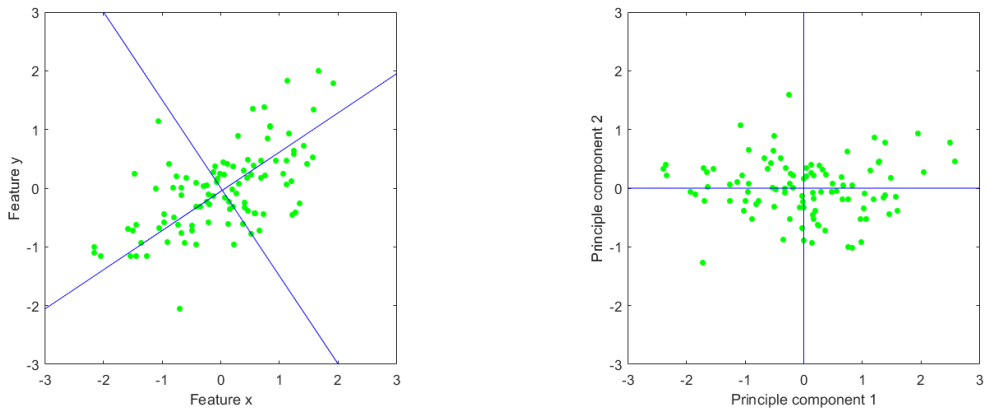


Figure 3.4: Principal component analysis with two feature into two components. Left is the original data and both principal components directions drawn. These lines become the axis in the right plot.

PCA can be extended to find a non-linear subspaces with high variance. Kernel PCA, for example, replaces or extends the features with a number of non-linear

features before normal PCA is applied. Another extension method is the principal curves approach in which the first component is represented as curve instead of a straight line, in such a way that the squared distances of the points to this curve is minimal [27].

PCA preserves distance information for both small and large distances. For many application this is useful and sufficient but some data structures, such as a spiral 3D distribution, require another approach. Points that are close to each other in Euclidean distance (the blue line in figure 3.5) could actually be far if you consider the overall structure of the dataset. For this problem, other dimension reduction techniques such as t-SNE work better.

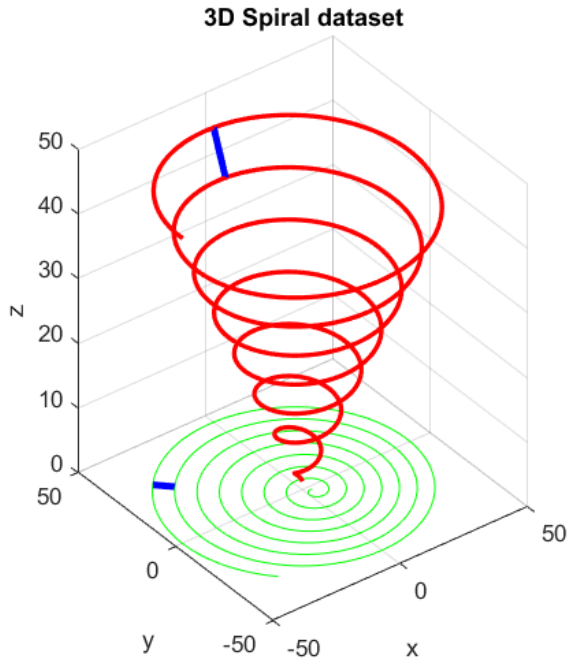


Figure 3.5: A dataset containing values in a spiral shape when plotted. For this situation, euclidean distance as equality measurement is not a good method.

3.3 Outlier Detection Methods

AD can be categorized into explicit detection and deviation methods [28]. The first group of detectors would fall within the group of supervised learning algorithms but as mention in section 3.1.2, most of the input data will be unlabelled, therefore unsupervised approaches are more common for AD. Moreover, for most anomalies it will

not be possible to define these abnormal values without any information about the non-anomalous or normal instances, mostly because it is unclear what an anomaly actually is. New types of rare events will therefore be easier to detect with unsupervised methods [29].

A variety of unsupervised machine learning techniques for AD will be evaluated from the basics to more advanced methods to find ways to detect the different possible anomaly types. Many surveys cover most of the following methods as well, but lots of variations and alternatives could be interesting for any form of AD [11], [17], [30], [31].

The different types of unsupervised detectors are [29]:

1. **Statistical methods**:
   The goal of statistical analysis is to find a Probability Density Function (PDF) f for which $f(x)$ is large when instance x is normal and $f(x)$ small when it is an anomaly. By using a threshold ϵ we can define x as anomaly when $f(x) < \epsilon$.

2. **Distance based methods**
   For this group of methods, outliers are detected by comparing the distances among instances or clusters [11], [29].

3. **Profiling methods**
   The profiling methods try go get an idea of what normal behaviour is for the specific instances. Sudden unexpected changes of feature values are reasons to flag this instance as anomaly [32].

4. **Model based methods**
   Model based approaches detects an outlier when the instance does not agree with a calculated model, which is generated on the normal data.

Outlier detection methods often use more than one of these principles and are therefore not simply considered one type of detector. Hereafter a series of commonly used as well as less known outlier detection methods are given and explained how they detect anomalies.

3.3.1 Statistical methods

Box Plot

The box plot is one of the simplest statistical techniques used for AD. Univariate or multivariate anomalies are indicated as such when they exceed the min or max anomaly limits (whiskers) which are defined as 1.5 times the Inner Quartile Range (IQR) and will contain about 99.3% of the values [11]. An example of a box plot can be seen in figure 3.6.

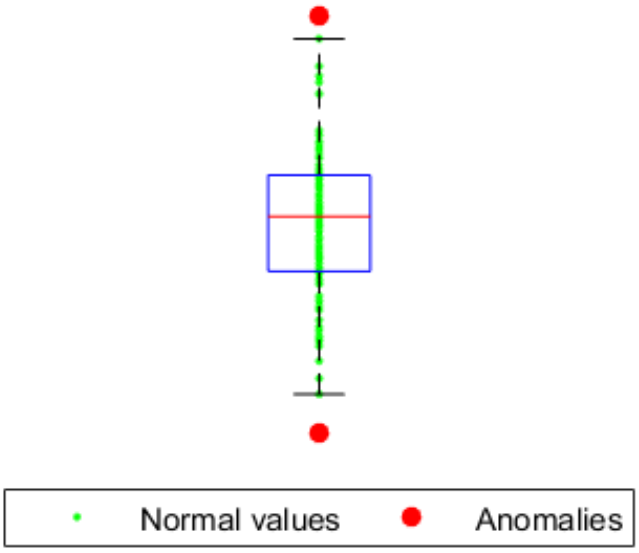


Figure 3.6: Box plot representation of the dataset with normal data within the two whiskers and two instances marked as anomalies. One above the top, and one below the bottom whisker.

Gaussian (Normal) distribution

Another common statistical technique to detect outliers is by defining a Gaussian distribution for one or multiple features. This technique assumes a normal distribution of the given features and calculates the parameters of those normal distributions by using Maximum Likelihood Estimation (MLE). Any instance with feature values outside of the expected range is marked as anomaly.

The distribution of a Gaussian distributed feature in the dataset is modelled as

$f(x; \mathcal{N}(\mu, \sigma^2))$. A visualization of the AD is visible in figure 3.7.

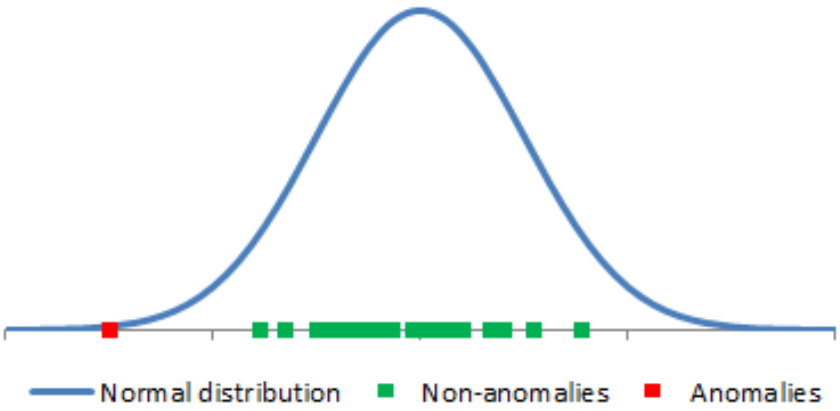


Figure 3.7: Gaussian normal distribution

Gaussian Mixture Models When this data is not evenly distributed around one value, but around two or more instead, a Gaussian Mixture Model (GMM) can be used [33]. These are combinations of normal distributions with each have a weight factor, summing up to 1 [34].

Multivariate Gaussian model If a correlation between features is not obvious or known but highly probable, a multivariate Gaussian model can be used [35]. Instead of a distribution for each feature it will create one model for a combination of features, having a μ vector containing the averages of all features and an $N \times N$ matrix with the variances for all combinations of features. The resulting PDF for $\mu = \binom{0}{0}$ and $\Sigma = \left(\begin{smallmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{smallmatrix}\right)$ is shown in Figure 3.8.

The drawbacks of this method are the computational power needed for the variance matrix and the size required for the training set whenever the feature space is big.

Histogram

An often used non-parametric statistical method is the histogram [36]. The feature values are split in buckets either by category or by dividing the range in equal parts.

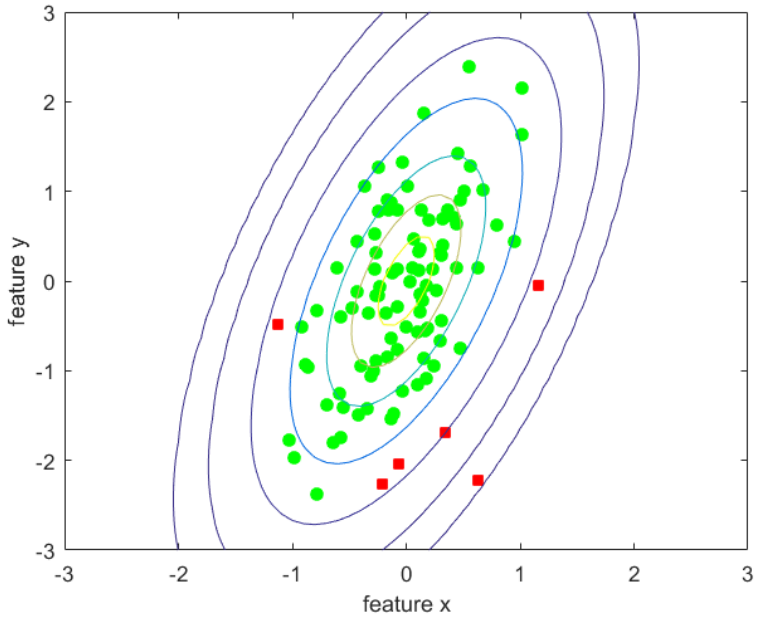


Figure 3.8: Multivariate Gaussian of two features with a probability threshold of 0.02

Subsequently, a histogram is generated based on the amount of data per bucket. A histogram can look like a Gaussian distribution if the data is normally distributed or look more like any Gaussian mixture model otherwise. It is possible to either use a histogram as one feature of an instance or use one for the complete dataset.

A probability of any value x can be calculated by counting the values that are in the same bin as x and divide this by the total amount of samples as follows:

$$\hat{f}(x) = \frac{1}{N \cdot h} \sum_{i=1}^{N} \sum_{j} I(x_i \in Bin_j) \cdot I(x \in Bin_j) \tag{3.1}$$

Where h is the bin size and $I(x \in Bin_j) = 1$ if $x \in Bin_j$.

Since this approach is highly depending on the bin size h, it would be preferable to make it as small as possible. This will better capture the data but will also create empty bins. A way to overcome this problem is making use of kernels.

Kernel Density Estimation (KDE)

A kernel function influence the area around every data point equally. Instead of looking at observations that fall into a small interval containing x, as we do in histograms,

KDE looks at observations falling into a small interval around x.

Because of its radial symmetrical and smooth function, a commonly used kernel $K(x)$ is the Gaussian kernel (as defined in equation 3.2) [37], [38].

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2) \tag{3.2}$$

The probability of any Kernel Density Function (KDF) $\hat{f}(x)$ is the sum of all the probabilities of the kernels at x:

$$\hat{f}_h(x) = \frac{1}{N \cdot h} \sum_{i=1}^{N} K(\frac{x - x_i}{h}). \tag{3.3}$$

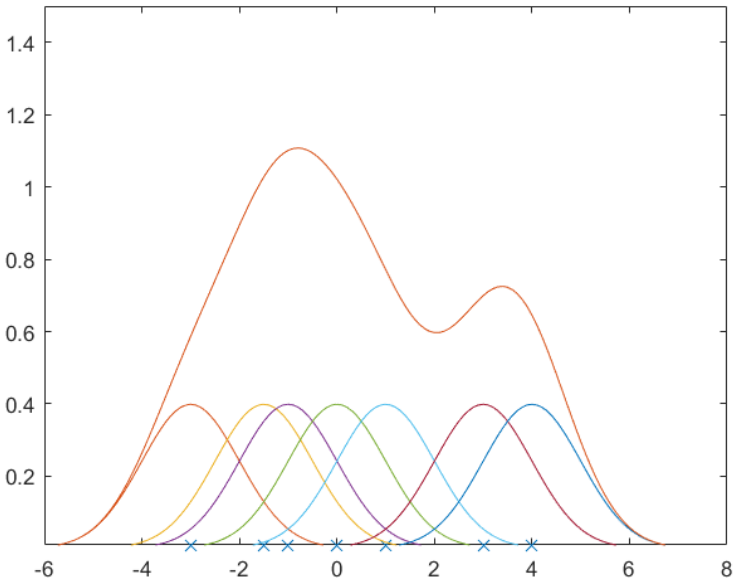


Figure 3.9: A visualization of the use of Gaussian kernels and the resulting density function

In this equation, h is the bandwidth of the kernel, a value specifying the distance a data point should have influence over the surrounding feature space. The smaller h, the better it captures local points but the data is more prone to over fitting.

Other kernels such as a uniform, triangle or Epanechnikov can also be used for the same purpose [38].

SVDD

The goal of Support Vector Data Descriptors (SVDDs) is to find a boundary around the dataset which encapsulates all normal values [39], [40] defined by the function $y = \theta_0 + \theta_1 k_1 + \theta_2 k_2 + \theta_3 k_3 \cdots \geq 0$, where k_i is a kernel with weight θ_i. A commonly used kernel is again the Gaussian for which the centre location μ is called a landmark. Data within the boundaries (for which $y \geq 0$) is considered normal and data outside ($y < 0$) an anomaly.

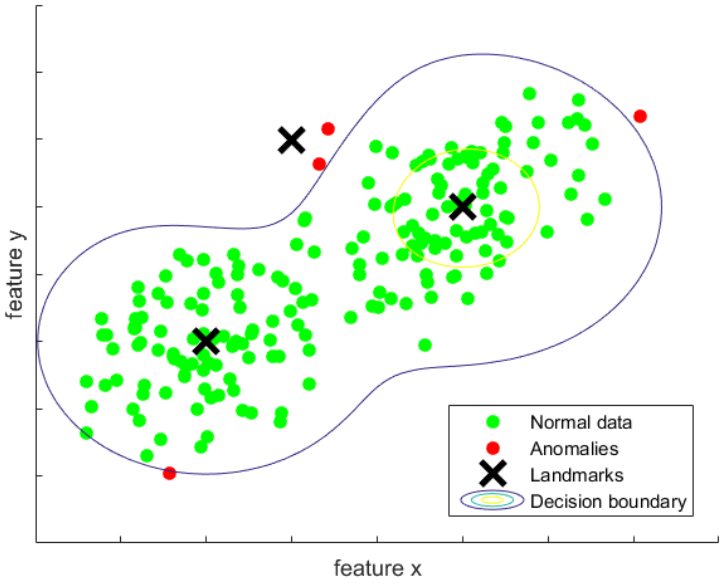


Figure 3.10: Visual representation of 3 landmarks, of which one falls outside and two within the decision boundary.

Whenever the landmarks are placed at all data points and all corresponding θ values are equal, this method us similar to KDE, mentioned in section 3.3.1. On the other hand, the ability to have different locations and weights for each (Gaussian) kernel is comparable to GMM. One important difference with those two methods is the ability to apply negative weights ($-\theta$) to kernels, which can make the corresponding landmark fall outside the boundary (see figure 3.10).

3.3.2 Distance based

K-Nearest Neighbors (K-NN)

A shortcoming of KDE arises when the densities of the cluster(s) vary. One fixed distance for each instance close to a dense cluster might still contain k instances although other instances inside this dense cluster would have significantly more than k instances within its neighbourhood.

In those cases, K-NN can be used as an alternative. Instead of looking at the number of neighbours within a fixed distance, this algorithm tries to find (k) closest neighbours creating a neighbourhood of exactly k instances. The distances to those neighbours, for example the distance to the furthest or the average distance to all instances in the neighbourhood will be compared to find outliers.

ARTMAP

An alternative clustering method to map normal data is Adaptive Resonance Theory (ART)MAP. When trained on non-anomalous data, it creates boxes encapsulating the data points. Whenever a new normal instance is received, it tries to stretch the box to capture this instance but when this data point is far away from other normal data, a new cluster (box) is created. Data points outside of the existing boxes are outliers based on this model.

Stochastic Outlier Selection (SOS)

SOS is a clustering structure method just like K-NN with the exception of the distance parameter. The neighbourhood of K-NN is based on an exact number (k) neighbours of an instance, SOS uses the distances to all instances to define the neighbourhood. The neighbourhood has not a strict boundary as K-NN but is the variance σ in a Gaussian distribution for which μ is the instance value [41].

Perplexity is a smooth measure for affective number of neighbours, comparable to the k value in K-NN.

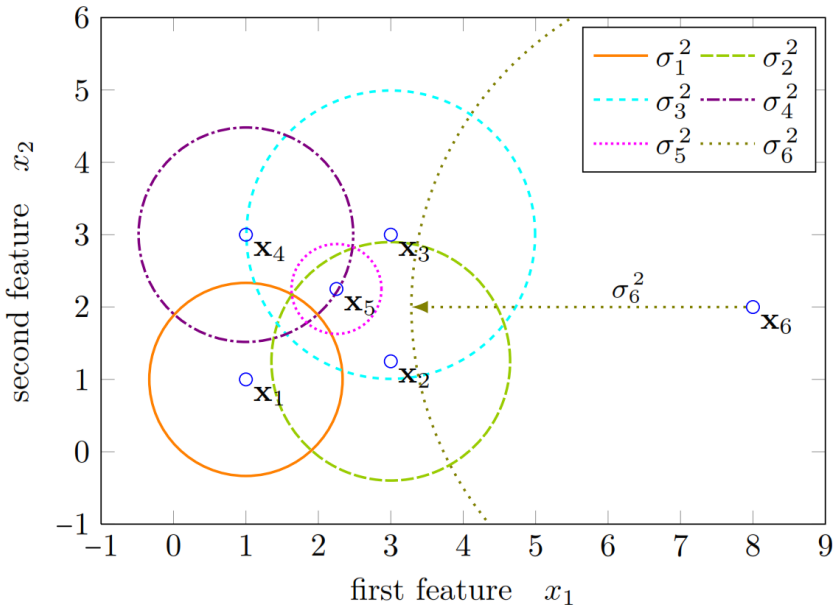


Figure 3.11: The variances σ_i^2 generates the same number of neighbours (perplexity) for every instances x_i with a perplexity of 3.5. Fig. from [41]. The circles are not borders like K-NN but an indication of the variance of the Gaussian at this point.

Dissimilarity The distances between all points forms the dissimilarity matrix. Usually these are Euclidean distances but as for the other methods, this can be any dissimilarity function. For Euclidean distances this matrix will be symmetric, so the distance $d(i,j) = d(j,i) = ||i - j||$ for all instances i and j but this does not have to be true for any dissimilarity function.

Affinity Affinity and dissimilarity are in some way opposites. The bigger the dissimilarity the less affinity the instances have. The affinity a_{ij} of two different points i and j is defined as:

$$a_{ij} = \exp(-d_{ij}^2/2\sigma_i^2), \tag{3.4}$$

in which the σ_i is the boundary for instance i. Due to this instance specific variance, the affinity matrix is not symmetric any more, so the affinity of i towards j does not have to be equal to the affinity of j to i. The diagonals are for both the dissimilarity as for the affinity matrix defined as 0. So instances has no affinity with itself.

Binding probability The affinity matrix is not a probability distribution because the affinities from an instance to all others do not add up to 1. Whenever we normalize every row, and make it a probability distribution we will get a binding matrix. This

terminology is based on the Stochastic Neighbor Graph (SNG) theory where two vertices are connected by directed edges based on probabilities generated by the affinity.

$$b_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{ik}}, \tag{3.5}$$

where i is the row and j, k the columns in the affinity matrix.

Any SNG can be generated in which every node (data point) binds to exactly one other node based on the binding probabilities. Nodes can have more than one vertex connected, which makes this node a neighbour for those instances. Any node with an in-degree of 0 is nobody's neighbour and therefore an outlier based on this SNG.

The number of times any instance x_i is considered an outlier and the probability of the graph $g \in G$ in which this instance is an outlier determines the outlier factor for this instance:

$$p(x_i \in C_{outlier}) = \sum_{g \in G} 1\{x_i \in C_{outlier} | g\} \cdot p(g) \tag{3.6}$$

Without looking at the SNG, we can determine this outlier factor directly from the binding probability matrix as follows:

$$p(x_i \in C_{outlier}) = \prod_{j \neq i} (1 - b_{ji}) \tag{3.7}$$

This equation looks at the columns of all instances, to see what the probability is of any other instance binding to it. When it has a high probability of being an outlier (higher than a certain threshold), we consider it one.

Local Outlier Factor (LOF)

The LOF algorithm compares the density of the instances with the average density of its nearest neighbours [29], [42]. As for SOS, the neighbourhood $N_k(p)$ of an instance p is defined by a distance $k\text{-}distance(p)$ around the instance encapsulating minimal k other instances (see figure 3.12).

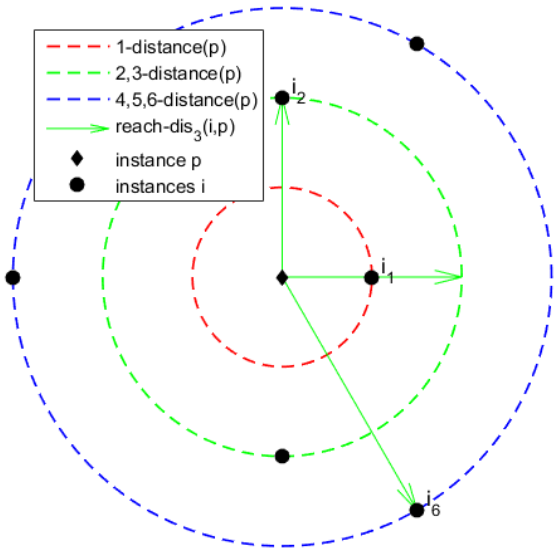


Figure 3.12: All possible $k\text{-}distances$ for an instance p and the $reach\text{-}dist_3(i,p)$ for i_1, i_2 and i_6. The distance $d(p,i_1) < 3\text{-}distance(p)$ so the $reach\text{-}dist_3(i_1,p) = 3\text{-}distance(p)$ but $d(p,i_6) > 3\text{-}distnace(p)$ so the $reach\text{-}dist_3(i_6,p) = d(p,i_6)$.

The reachable distance $reach\text{-}dist_k(i,p)$ of instances i,p is equal to the $k\text{-}distance(p)$ if the instance i is within the neighbourhood of p or the distance from p to i otherwise:

$$reach\text{-}dist_k(i,p) = \max \begin{cases} k\text{-}distance(p) \\ d(p,i) \end{cases} \tag{3.8}$$

Some objects in the neighbourhood of p will not have p in their neighbourhood, since the boundary ($k\text{-}distance$) is individually determined. An instance p has a high local reachability density if the object inside the neighbourhood of p have p in their neighbourhood as well. The local reachability density of an object p is defined as [42]

$$lrd_k(p) = 1/\frac{\sum_{j \in N_k(p)} reach\text{-}dist_k(p,j)}{|N_k(p)|} \tag{3.9}$$

As equation 3.9 shows, the $ldr_k(p)$ is based on the average $reach\text{-}dist$ of all objects inside the neighbourhood of p to p.

The outlier factor is a comparison between the reachability of p and all the in-

stances in the neighbourhood of p as is defined as follows:

$$LOF_k(p) = \frac{\sum_{j \in N_k(p)} \frac{lrd_k(j)}{lrd_k(p)}}{|N_k(p)|} \tag{3.10}$$

K-Means Clustering

There are 5 different types of clustering (Partitional, Hierarchical, Grid-based, Model-based and Density-based [**?**]) of which Partitional K-means clustering can be seen as one of the best known classical principles [43].

K-means iteratively determines what the centroids of the clusters should be given a value K as the number of clusters. Following a two step process where it first assigns all instances to the randomly placed centroids. In the second step it places these centroids in the centre of the instances that are now part of that cluster. Consequently the procedure is repeated with the new locations of the centroids until they reached a steady location [44]–[46].

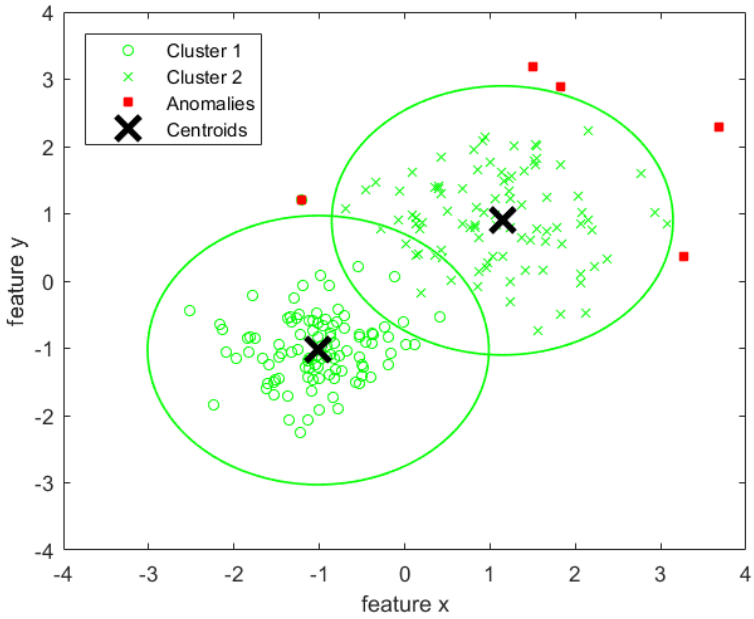


Figure 3.13: K-means cluster analysis with two features, two clusters and a distance threshold of 2. Whenever an instance does is not within a distance 2 from a centroid, it is marked as anomaly.

The algorithm requires a value of k to be chosen *a priori*, which should not be a

problem when the expected number of clusters are known and constant. However, in other cases an extra step of determining k is required. In [45] different techniques to evaluate the clusters are mentioned how to calculate how well the clusters are separated.

Silhouette coefficient This evaluation method averaging the silhouette values $s(i)$ for all instances after the k-means algorithm divided the data into k clusters. The first is to calculate for every instance in a cluster $i \in A$ the average distance to all other instances of the same cluster.

$$d(i) = \frac{1}{|A|} \sum_{\forall j \in A} d(i, j)$$

This will be subtracted from the average distance to the points in its nearest other cluster B and divided by either the max average distance to instances in A or B, which will usually be B otherwise it might be in the wrong cluster.

$$s(i) = \frac{b(i) - a(i)}{\max b(i), a(i)}$$

where

$$b(i) = \frac{1}{|B|} \sum_{\forall jinB} d(i, j)$$

The bigger $s(i)$ and subsequently the average over all silhouette coefficients, the better divided the clusters are.

3.3.3 Profiling based

Trajectory clustering

So far we did not explicitly take time into account, except through features. Trajectory analysis uses historical information of an instance as input for AD.

Some of the previously mentioned methods can perform well for trajectories analysis, when the whole trajectory is considered a feature. For example Discrete Spacial Distribution Representation (DSDR), which maps the paths as 2D features on a plane, generating probability distributions of a trajectory.

The Isolation Forest (iForest) method mentioned in [47] explains another trajectory anomaly detector. It compares a sequence of key points in routes and marks an instance as anomaly when this sequence has a different order or set of key points. A graph representation of the sequences can also be used to detect cyclic behaviour.

Association rules

This method tries to find a relation between features. Whenever such association exists, it is likely for B to happen if A happened if A is associated with B. These events can occur simultaneously or in a later moment in time. Associations can also form with more features, for example if A and B occur, there is a high probability of C occurring some time later.

Peer Group Analysis (PGA)

PGA is an example of a profiling method where a model is fit to what is considered a normal pattern for individual instances over a fixed time period [32]. This method compares the individual trends to its peers, which are other instances considered similar to the instance.

3.3.4 Model based

Linear regression

A linear regression model uses the concept of PCA and one of the statistical methods mentioned before. it tries to fit a line to the data points which would be the first principle component in PCA. The deviation of this line is considered the anomaly factor. More complex lines such as polynomial curves can be used as well.

Artificial Neural Networks (ANNs)

Neural networks are applicable for both supervised and unsupervised applications. Commonly used feed-forward and feedback networks are part the first group be-

cause those are able to adapt based on the errors produced by the network but require labels to determine whether an instance is correctly classified. However, for a third category of ANNs called competitive or self-organizing, there is no *a priori* set of labels needed and therefore usable in unsupervised settings [48], [49].

The idea of a neural network is to mimic the brain, therefore the nodes are often called neurons and the connections between them synapses. An ANN has each layer of the network fully connected to the previous layer, so every neuron of the first layer has synapses to all input features as can be seen in figure 3.14.

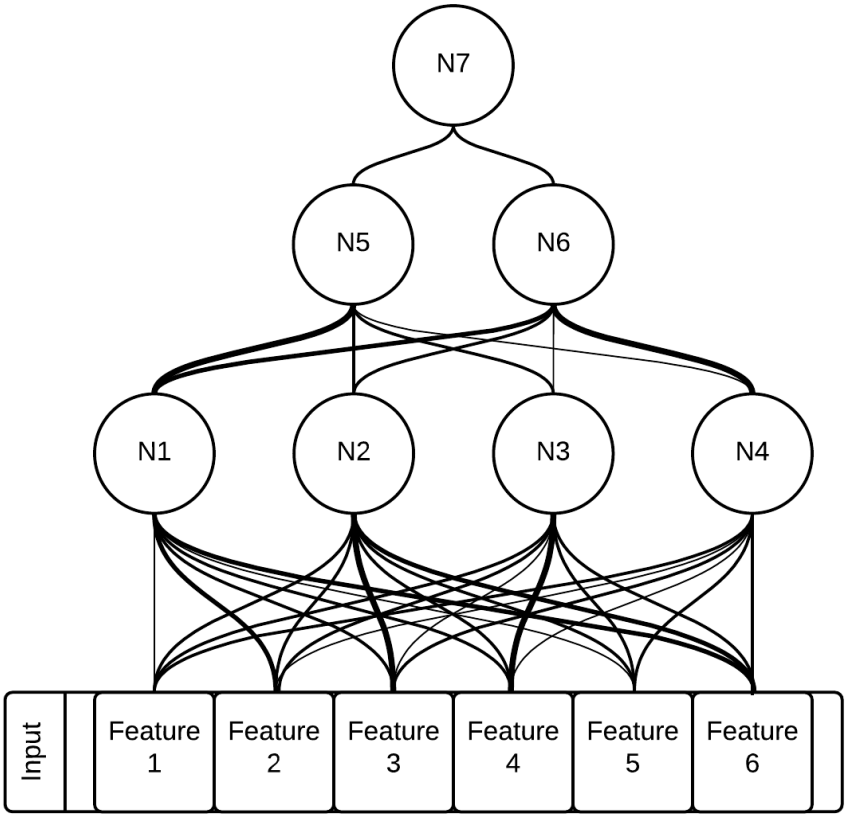


Figure 3.14: An example of a multi-layer Neural Network with two hidden layers and one output node

Some ANNs can use the same mechanisms as other AD methods. For example in [49, p. 84], the Pattern Associator Paradigm is described to find a set of input patterns to find output patterns just like association rules do or can perform PCA with linear Neural networks.

Part of the learning principles of a network is to adapt the weights of the connections thus the strength of the synapses. One way to do this is according to

the ”Winner takes all” mechanism as competitive learning. The idea of this is to strengthen the synapses of the neuron that matches the input the most. This way any new input closely related to the current will again generate the same ’winner’ neuron.

Self Organizing Maps (SOMs) A Self Organizing Map (SOM) is such a generic technique that uses the winner takes it all principle to associate a number of neurons to certain clusters in the high dimensional data. However, in this model the winning neurons also move other neighbouring neurons into the direction of the input value [27], [48]. The neurons are therefore interconnected, which generates a neighbourhood around the winner neuron.

Hierarchical Temporal Memory (HTM)

HTM is, like ANN a learning method based on a model of the human brain. Where ANN is a more mathematical approach, HTM tries to mimic the brain even more [50].

The model exists of a number of columns, all having connections to a random subset (instead of all) of the input bits. These ’synapses’ have weights as well but can either be strong or weekly connected depending on a threshold (see figure 3.15).

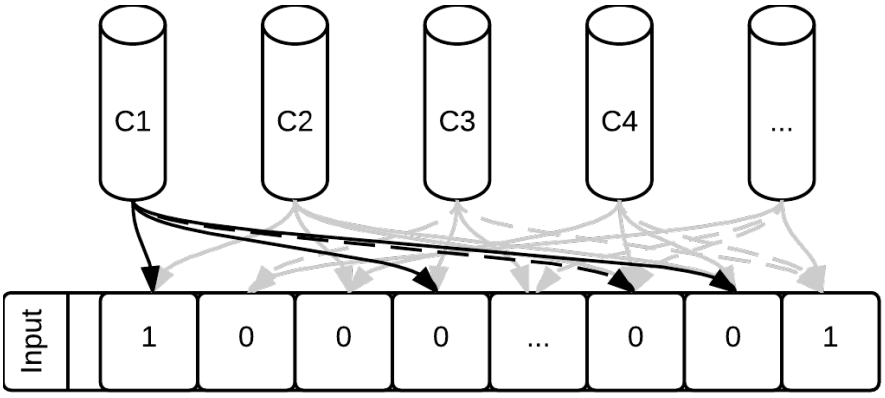


Figure 3.15: Representation of the relation between columns and input.

Each column contains a certain amount of cells. Those cells can be in three different states a any moment in time: inactive, active or predicted. Cells are mutually connected to a subset of cells around them.

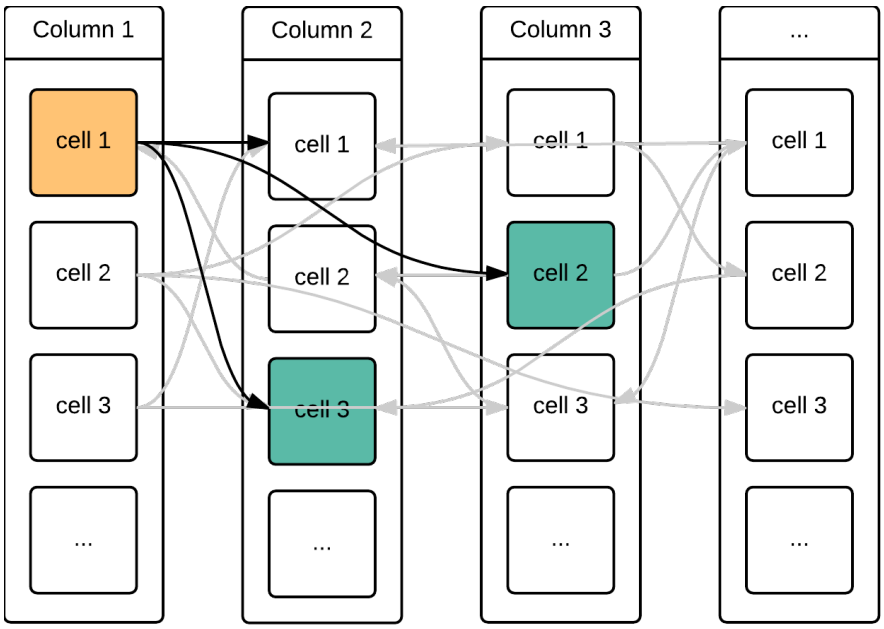


Figure 3.16: Representation of the relation between cells within the columns with two active and one predicted cell

Encoding The first step in the algorithm is to create a binary representation of the input data. The data should be coherent in similarity and therefore should not contain information in least or most significant bits. For example, a scalar encoder[1] would represent the number 7 as 111000000000 instead of its binary representation 00000111. The encoder creates a number of buckets based on the $\min$ and $\max$ values for this feature. The previous example has a range from 0 to 100 and 10 buckets (111000000000, 011100000000, 001110000000, ..., 000000000111). This will put values $0-9$ in the first bucket, $10-19$ in the second, etc.

Spatial pooling Whenever a new input is present, every column will be scored according to the number of connected synapses (the weight of the synapse $\geq$ threshold) with an active bit. Next, a subset of columns with the highest scores will be taken and these are now considered the active columns generated by this input.

For learning purposes all weights of the synapses connected to these columns will be increased when they had an active bit or decreased when they were linked (either strong or weak) to a 0-bit.

[1] https://github.com/numenta/nupic/wiki/Encoders

Temporal pooling For the next step we will look at the cells within the active columns and consider two possible options. Either one of the cells is currently predicted or none of the cells is. When a cell is predicted it should become active, otherwise all cells within the column will, which is called bursting.

Every cell can set its state predicted based on the cells it is connected to. Whenever it becomes active at a certain moment t, it makes connections to neighbouring cells that were active at $t - 1$.

A bursting column is an anomaly indicator since this is not a predicted active column. The more columns are bursting at the same time, the more likely this input is an anomaly. Therefore, the anomaly score is the ratio between the active and the bursting columns.

3.3.5 Combinations of methods

Most outlier detectors use a combination of technique which works best for the input data they have to use and the output they want to generate. For the same reason there is no single best solution and lots of variations on the previously mentioned methods are possible.

Outlier likelihood

Whenever any AD is detecting (false) anomalies on a frequent time interval, this can be considered as normal behaviour of the detector. A separate AD can be trained on this interval of detected outlier as it were a cascade or chain of ADs. The detector will basically mark an anomaly whenever this happens on an unpredicted temporal or spacial location.

3.4 Abnormal Behavior Detection

To understand what kind of anomalies the outlier detection methods mentioned in the previous section can detect, we first have to look at possible types of anomalies within the context of surveillance as well as other research in detecting those

anomalies.

3.4.1 Definitions of anomalies in surveillance

Even with little information in terms of features, there is a range of possibilities to detect outliers. Some of them can be defined in advance, these are usually the anomalies where an operator will be looking for. Others might be invisible or hard to detect by the human eye, but can be important nevertheless.

Speed

Speed is one of the first anomalies that come to mind. It can either be a point anomaly when an object has an unexpected high speed or several contextual anomalies, such as a pedestrian walking on the road or a bicycle riding on the pavement [51].

Direction / Course

The direction of an instance can also be subject to AD. Usually this will contain a contextual aspect as well, for example moving in the opposite direction of everybody in the neighbourhood or moving towards an object of interest.

Positions and Paths

If one instance is following a path which does not fall within the normal tracks (generated by non-anomalous instances) it should be considered anomalous [51], [52]. These situations can either be a local point anomaly, for example when somebody's position does not occur in any of the normal paths like walking on the grass, or a global anomaly when an instance makes an illegal turn.

When some instance takes a combination of paths which creates not the fastest, easiest or most logical route, this instance is an anomaly [47]. This includes circling a certain building or object.

Collective anomalies

If a certain amount of instances suddenly tend to go to the same place or in the same direction, there might be something worth looking at. The exact opposite, when everybody is trying to avoid a certain area is also suspicious group behaviour. Other anomalies within this category are interactions between instances and the detection of pick pocketing, snatching or pursuing [53].

3.4.2 Detection Methods

These are examples of how other researches did anomaly detection in human behaviour. Most of them can be categorized into two groups: one group of detectors using video images to capture behaviour on one crossing or square. Others try to do anomaly detection for a wider area such as a whole city, port or waterways.

Video surveillance Different preprocessing steps are used to detect an object in the image. This image processing step will be kept out of scope, instead we focus on the methods used when the images are transformed into objects or targets and with features such as the position in the image.

In [54], the motion patterns of tracked vehicles in the scene are learned from image sequences. The probability of an observed trajectory matching the learned motion patterns determine the abnormality score. The probability is based on a combination of trajectory clustering and a chain of Gaussian distributions.

As for most tracking systems, [51] produces a set of tracks based on observations where each observation is a set of features such as position and time stamp. This detector computes transition vectors from one observation to a number of future observations in the same track. These vectors are used to calculate a PDF as a multivariate GMM for every location in the scene. This PDF estimates the probability of observing an object with these features (speed, direction, etc) at this location in the scene.

Flow vectors are used in [22] for representation of trajectories. These are transformed into prototype vectors using VQ (see section 3.2.1) and a PDF of the distribution of points associated with this specific prototype vector. A preprocessing step of re-sampling is required to prevent vectors to be densely distributed when the

speed is low and sparsely when the object travels with a higher speed.

For better context analysis, it can be useful to generate semantic regions within the scene. For example detecting different road segments such as pedestrian crossings, traffic lights or other stopping areas, one-way streets etc. This semantic model can be obtained by clustering trajectories based om some class (e.g. pedestrians and cars), spacial distance and directions [55].

Motion patterns are also used in [56] to generate pattern models. They use clustering algorithms to group different the motion patterns to reduce the number of models by generating one model for every cluster.

The State-Based Anomaly detection by [53] requires features to be converted into classes as a method to compare them. Speedstate (stopped, slow, medium, high), CourseState (e.g. north, north-east, east,) are feature used for their detection method. The categorical feature RelationState (inFront, behind, left, . . .) which is an indication of their relation to the closest neighbour, is useful in detecting collective anomalies. The actual detection is performed by a table lookup, checking if a new state is in the normal set.

In [57], cars are tracked while crossing an intersection and their trajectories analysed for anomalies. The trajectories are preprocessed to remove errors from the tracking first, followed by clustering to find template trajectories (which are the cluster centroids). A Gaussian distribution is calculated for each template trajectory and trajectories with a higher speed than the mean plus standard deviation are considered anomalous. Furthermore trajectories are anomalous when they are not comparable to any of the template trajectory.

Wide area surveillance The Automated Anomaly Detection Processor (AARP) mentioned in [14] is designed to detect anomalies of land, sea and air targets detected by Ground Moving Target indicator (GMTI) radar or sensors on Unmanned Aerial Vehicles (UAVs). Again a sequence of observations with each observation represented as vector containing the feature values and time of the observations is used as input for the detector. AARP uses a SOM to generate model vectors from the normal dataset. A GMM with the model vectors as means and the normal data around the vector as variance is used to be able to provide feedback on what feature triggered the anomaly detector.

An AD system for port SA is presented in [16] where a ARTMAP is used to

cluster features (for example the speed) of a vessel according to their type and the zone (open water, inner water, dock perimeter or dock) they are currently in. The boundaries of each cluster can stretch whenever a new data is added. If new data is still outside of the boundaries, it is considered an anomaly.

A method of motion anomaly detection in ports and waterways is also proposed in [58]. The information acquired by Automatic Identification System (AIS) messages such as name, type, size, position, speed, etc. can are used as features. The KDE used in this method is contextually based on where the ship departed. Trajectories that departed from the same origin share the same model, where all measured locations within the trajectories are used in the KDE.

Two different algorithms for AD are covered in [52]: Their research evaluated GMM and KDE on comparable AIS data, which includes a wide range of features. For the detectors the position (latitude and longitude) and speed vector (also split into latitude and longitude) are used as features and to eliminate contextual problems they only use cargo or tanker vessels in the dataset. Both methods used showed a high rate of false negatives, most likely due to the use of a grid structure. The densities at the border of each cell are low which is counterproductive when trying to construct PDFs for each cell individually.

AIS data of tracked vessels was again used in [34], where a grid of the area is constructed and GMMs are used for every cell. This time, each cell has two normal models, one univariate ('base') model to captures the velocities in that grid cell and a second ('extended') bivariate model with the velocity and spacial positions to capture the correlation between speed and position. Instead of using the regular EM algorithm to find a predefined number of Gaussians, they use a greedy extension which will also find how many Gaussians are needed.

In [59], [60], AIS data is used as input for ARTMAP classifiers. The correlation between the speed and the location were used as features. However, instead of the exact latitude and longitude positions, the locations were grouped into classes ('port', 'open water' or buoy numbers).

3.4.3 Conclusion

Even with the list of possible anomaly detection methods mentioned in section 3.3, it is remarkable to notice how most of the implementations and research on surveil-

lance only use a small subset of those methods. One reason for this might be the limited amount of features (dimensions) available and used for those anomaly detectors. On the other hand, these methods might also be working sufficiently for the proof of concept in those researches.

3.5 Evaluation

Whenever the data can be represented in a low dimensional feature space, visualizing the data could help to get an idea of what instances are classified as anomalies and whether these are in fact valid. An expert in the domain of the data could be consulted for this task [61]. Both true and false positives can be evaluated using this approach and since the amount of anomalies should be smaller significantly than the dataset (otherwise it is not an anomaly), this should be feasible. However, true or false negatives are harder to detect since those instances can be hidden within the large number of instances.

The ratio of anomalies versus normal data is an indicator as well. If half the dataset is classified as anomaly, the detector seems to be less powerful. Knowledge about the data can be used to score detectors based on the expected ratio.

Some detectors can be evaluated by assessing their parameters generated by the training data. A Gaussian distribution with an unexpected high variance would indicate another value distribution of this feature. The same way a clustering algorithm could be evaluated by comparing the intra-cluster variance with the inter-cluster variance.

Chapter 4

Method

Literature and research on related work has given us several techniques useable for anomaly detection within and outside of the domain of this research. Most of the techniques require knowledge about the characteristics or are specifically made to detect one anomaly type. This research does not focus on one single anomaly type but rather tries to find a method to detect any given event in the dataset. A prototype capable of detecting different anomaly types using the techniques mentioned earlier is designed for testing and evaluation purposes, but can also be used as anomaly detection tool outside the scope of this research.

Due to the lack of information about the dataset and the anomaly types that should be detectable, it is not possible to find a technique by analysing the problem characteristics as explained in Figure 3.1 and Figure 3.2. Instead of designing a prototype to detect one specific anomaly type, we will aim to find and define an abstract detection technique, which is independent of what anomaly type it should detect. This technique will have different combinations of context, model and features available to construct a detector which can detect outliers based on a preselected combination. The anomalies this research will use for evaluation depends on the available dataset. Any of the offences mentioned in Chapter 2 which is present in the dataset and is the only event around the time of the offence, will be used as test event.

4.1 Prototype

The development of a framework for detecting anomalies in wide area data is done iteratively. The core of the system, such as the structure of the data, should be applicable for any detection method and is implemented first. In the following steps the different methods were added to the framework. A reason and advantage for designing the framework as a set of tools instead of one-method solution is to open the possibility to test the methods on different datasets containing other anomalies, which can require different methods and features.

4.2 Detection technique

Not all statistic outlier detection methods explained in Section 3.3 can and will be used during this research. A selection was made based on the success of the methods in other studies and the time available for integration of the method in the prototype. Based on these arguments, a Gaussian, GMM, histogram, Polynomial, and KDE are used as statistical models.

The data will be grouped based on a certain contextual aspect and for each group a statistical model will define what behaviour is normal for this group. Groups can be based on no context to find outliers based on everybody else, personal, which should be able to detect changes in personal behaviour and location based to compare people in the same area. More details about these contextual grouping can be found in Chapter 5.

The Features can be any characteristic of somebodies behaviour. Speed, direction and location are most often the focus in other and this research.

4.3 Simulator

The open-source Intent Driven Scenario Authoring (IDSA) simulator (Figure 4.1) used for evaluating the methods and design used in this research is provided by TNO. The simulation represents human behaviour based on the intentions and agenda each agent has.

Figure 4.1: The original IDSA simulator, used to simulate walking behaviour of people as well as (criminal) events.

Using a simulator as evaluation method has advantages and disadvantages. Behaviour can be designed according to specific requirements with a simulator but we can not assume this behaviour is realistically modelling real life behaviour. By using a simulator to generate the dataset we eventually make a detector reflect the behavioural model of the simulator and not the real world behaviour (see Figure 4.2). Data generated by observations of real world behaviour would be preferred but unfortunately unavailable.

The designed anomaly detectors are tested by how well they fit the situation as well as the ability to detect special anomalous events added to the simulation. These tests are based on recordings of 4 (simulated) hours. One of the test cases has no events at all and should contain only normal, non-anomalous data. The others test cases have anomalous events occurring at a fixed time interval.

4.3.1 Normal data

Purely normal data, which is a simulation without any generated event, is used to evaluate the models. The less outliers a model has, the less False Positives (FPs) it will generate (see Table 4.1 and Figure 4.3). On the other hand, it should still be able to detect the anomalous events thus having a high precision (which is the amount of

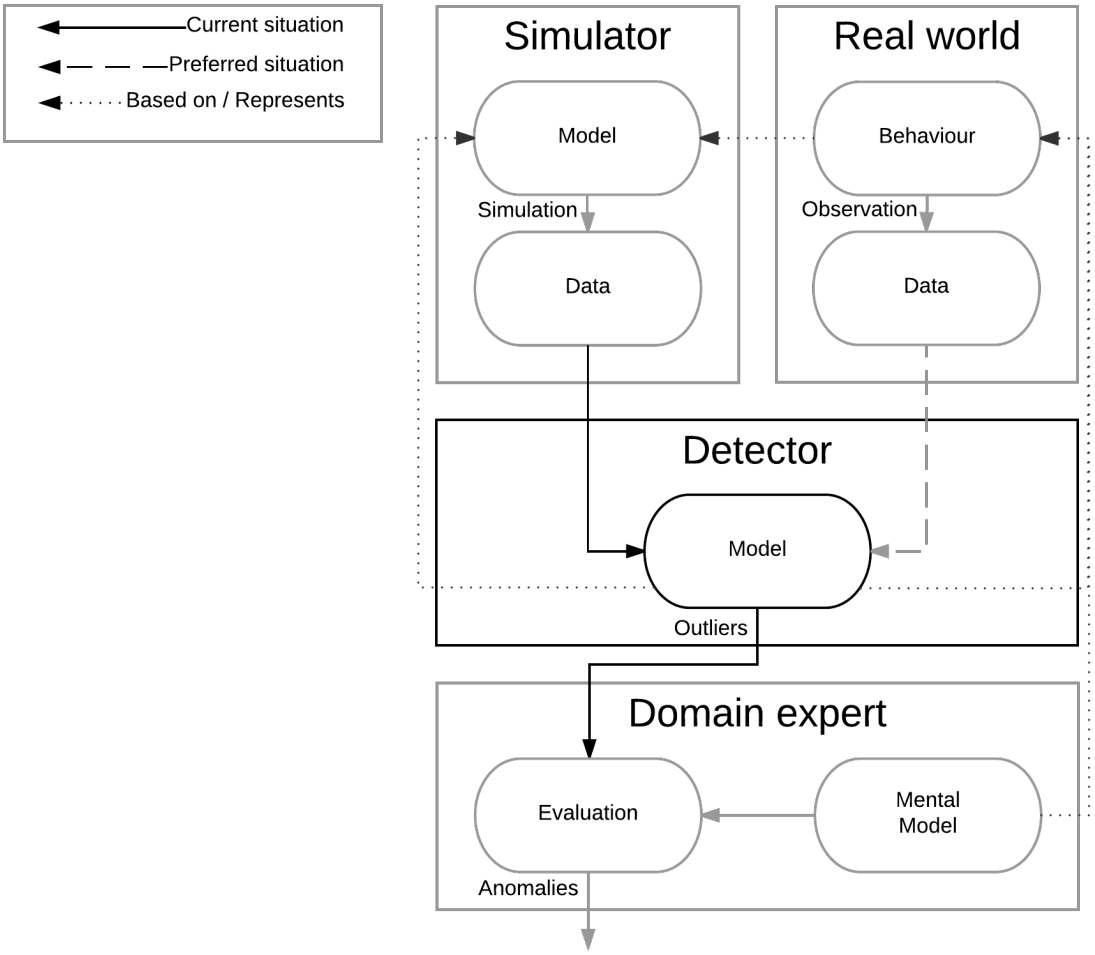


Figure 4.2: Evaluation is done using a simulator which is modelled to simulate real life behaviour. Training the detector on the simulated data will create a representation of the simulation model. In an ideal situation the data fed to the simulator is observed real world data, which will make the detector model actual behaviour.

Table 4.1: Confusion matrix of the relation between outliers and anomalies

	Outlier	Inlier
Anomaly	TP	FN
Normal	FP	TN

correctly detected anomalies divided by the total number of outliers found). A model with no FP is highly unlikely but not impossible. Inspection of the actual distribution of feature values and the PDF of the resulting model can provide insight in why a model has or does not have FPs.

4.3.2 Simulated events

A well fitted model is important for proper anomaly detection to reduce the number of false positives but it is even more important to find the real anomalous events. Testing what methods are able to find the simulated events required modifications to the simulator. Details about those changes can be found in Appendix B. It was possible to manually add events at certain times already, but automatic extraction of what agents in the simulator were involved in the incidents was not available outside of the simulator.

For the tests, 4 events are generated, one every hour (12:00,13:00, 14:00 and 15:00.). These events are planned to occur 37 minutes later (at 12:37, 13:37, 14:37 and 15:37) at different locations. This will give the participating agents enough time to arrive at the scene.

An increased number of detected anomalies at the times of the events suggests successful detection of the anomalous people but detailed analysis on what person is marked as anomaly is needed. The tests main goal is to evaluate the amount of True Positives (TPs) and False Negatives (FNs) (recall), since those are related to the detection of anomalies. Minimizing the number of FNs is more important than having a few FPs, especially when an operator can easily notice the difference.

Some of the detectors only have information about the location of a collective anomaly, not what agents were involved. To test the accuracy of those detectors, the location(s) of the found anomalies is compared to the actual start of the anomaly. False positives will increase the average distance so the smaller the distance, the better the detector.

Since the behaviour of the test events are modelled in the simulator, we can

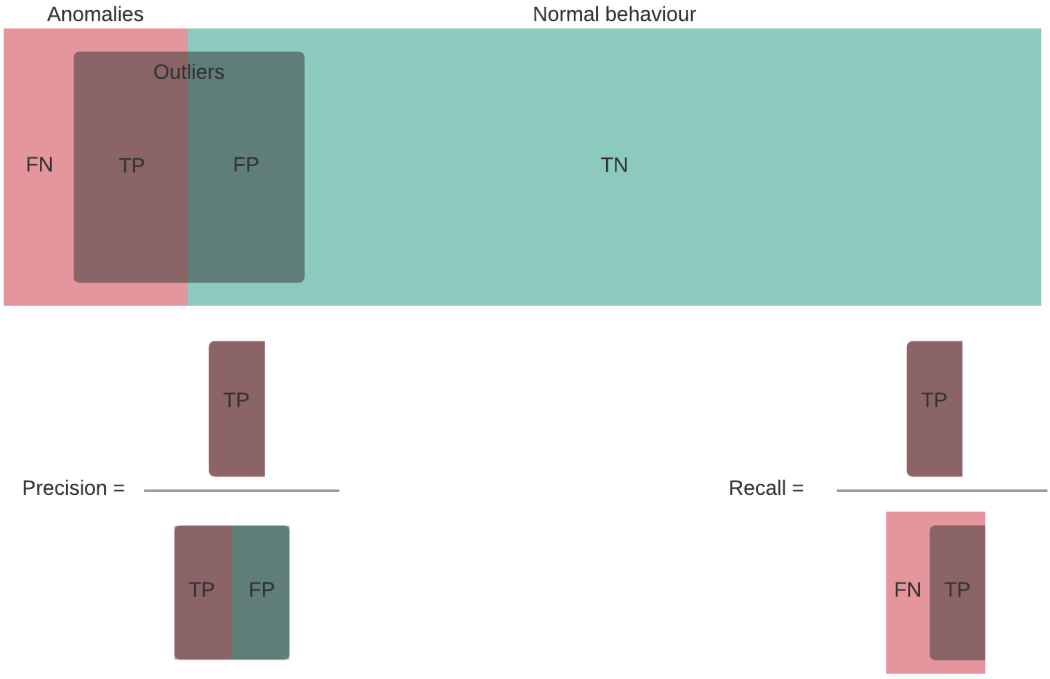


Figure 4.3: A visual representation of the relation between outliers and anomalies in terms of precision and recall.

easily see what behaviour we can expect during the event and compare these with the actual characteristics of these events in real life. Subsequently we can decide what detection methods are most likely to work to detect the event. Unfortunately, some of the events were not (fully) realistically modelled, therefore their behaviour is added to the simulation as described in Appendix B. For the same reason the three events mentioned below (procession, street and commercial robbery) are used for evaluation the detectors, models and designs.

Procession

A procession is modelled to have a set of agents following the same path. Each agent participating in this event will start with a small delay creating a chain of agents. The default procession has 30 participants, each starting 3 seconds after the previous agent. They will cover a distance of 500 meter, with an average speed of 3 km/h.

Pickpocketing, robbery and street theft

For the simulation, a street theft event has the offenders and victims walking to the crime scene, use either blitz or snatch methods to obtain the item and run back to where they came from. A default event has two victims and two offenders, all randomly selected at the moment this event is created.

Shoplifting and commerical robbery

The main difference between commercial robberies and street thefts as test method for this research is the location of the event. When the crime is committed inside a building and the offenders exit the building, no historical data about them is available since tracking was lost during this time.The default shoplifting event for the simulator has two offenders as well, but not a simulated victim because this person is inside during the robbery. For this reason the 4 event locations are also different from the procession and street theft.

Chapter 5

Prototype

In contrast to the detectors mentioned in literature, the designed system does not use strictly one method of detecting anomalies but is modular and easily customizable. The combination of detection method, model and feature opens opportunities to find anomalies that are not thought of before. Although not implemented, each combination can be used as a layer, determining what detector is more prominent in what area (see an illustration of this idea in Figure 5.1). This opens the ability to give priority to detecting running in the city centre where detection of public gatherings could be more important in suburbs.

Two types of anomaly detectors are implemented. The first type focuses on the predicted distributions of features (point anomalies) where the second type looks at the collective anomalies. Different detectors are implemented for both types, as can be seen in Figure 5.2. This section broadly describes design choices required to

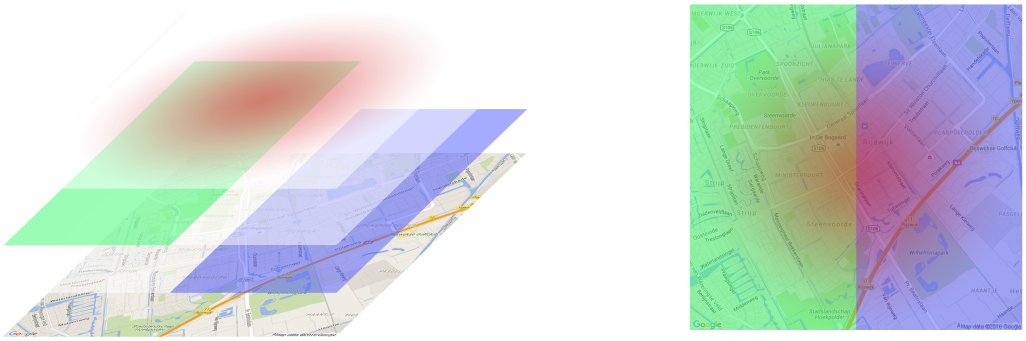


Figure 5.1: Prioritizing different detectors (represented as layers) in other areas and neighbourhoods.

understand how the detectors work. For more detailed information, diagrams and algorithms used and what other detectors are (partly) realized, see Appendix A.

5.1 Trajectory detectors

The trajectory detectors are commonly seen in previous research. They are able to perform point anomaly detection, mostly by statistical analysis. In the current trajectory detectors they are trained with one dimensional data. The actual values depends on the feature of the trajectory (speed, direction, etc).

5.1.1 Input

The detectors require spacial temporal input as four values namely an ID, X-position, Y-position and the time. After more than one position for a person is received, a segment between those points, which defines the movement of this person, is created. The sequence of these segments for one person form its path (see also Figure 5.3). A trajectory is defined as an abstract version of both a path and segment, having a start and end position as well as a start and end time. With these properties features such as speed and direction are deductible, which is the input for the trajectory detectors.

5.1.2 Detectors

The detectors each split the set of trajectories in different subsets, based on what context they use (see Figure 5.4). The detectors keep track of the models of all those subsets and adds new data to the correct model when provided. Anomaly checking is done by first determining to what context the given trajectory belongs and subsequently calculating if the feature value for the trajectory is an outlier according to the model PDF.

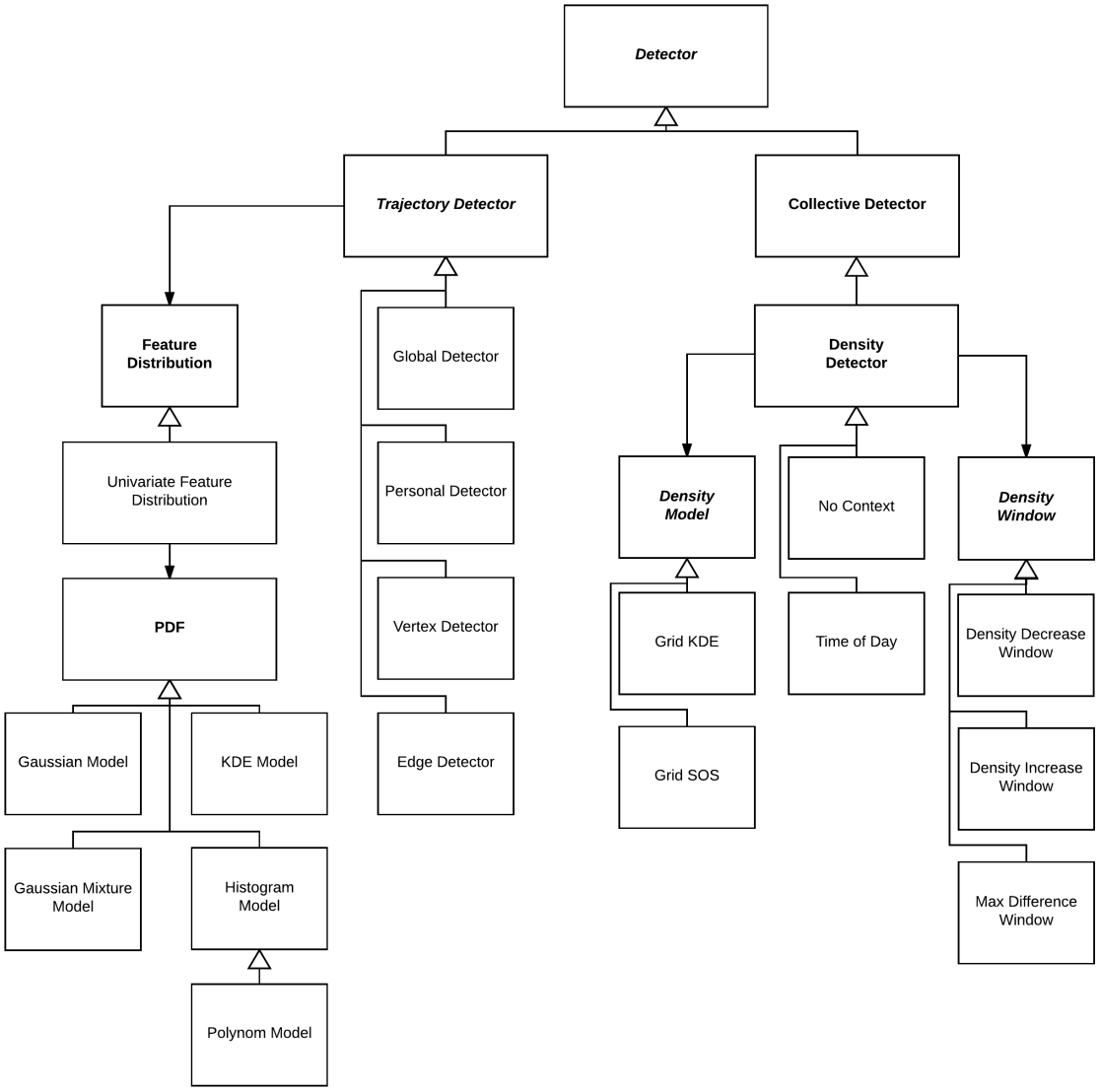


Figure 5.2: A diagram of the realized anomaly detectors, showing the relation between context (detector), feature/window and model (closed arrow) as well as the current possible implementations for these parts of the detector (open arrows).

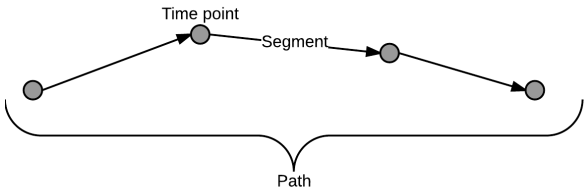


Figure 5.3: Definitions of time points, segments and paths.

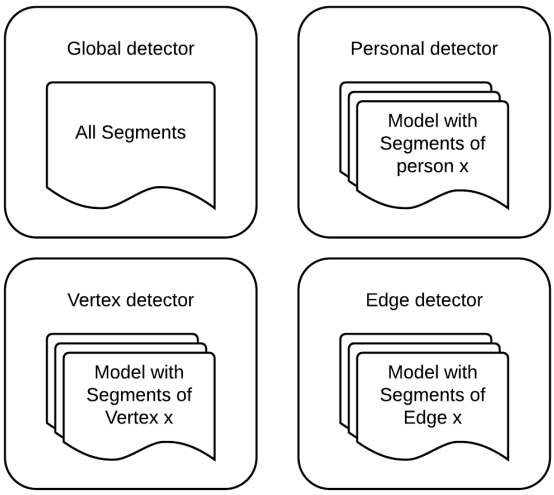


Figure 5.4: The detectors all have models of different subsets of the trajectories. The trajectories are divided into the subsets based on their context.

Global and personal detector

The global detector uses one of the statistical models to calculate the probability of the trajectory based on all (non-zero) trajectories received so far. Context such as location, time or who walked this trajectory are not relevant for this detector. The personal detector uses one model for each individual person making it contextually dependent on who the trajectory walked. It is therefore useful for detecting sudden changes in behaviour. The amount of available data per person is unfortunately limited and whenever somebody goes inside and the collected information is not usable any more.

Vertex & edge detector

An graph representation of the points and trajectories (called vertices and edges) are constructed to capture multiple trajectories at the same spacial location. These edges are comparable with the *model vectors* mentioned in Chapter 3 and represents the roads in the scene. Consequently an edge is the collection of trajectories with the same vertices as begin and end point. More details on how the graph is generated can be found in Appendix A.

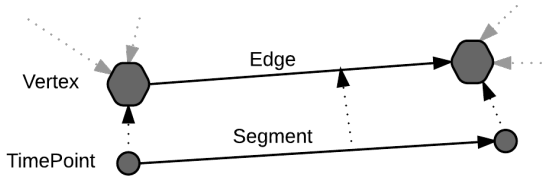


Figure 5.5: A visual representation of the mapping from trajectories to edges.

Trajectory type	Feature	Feature value
All Trajectories	Speed	Meter per second
All Trajectories	Angle	Radials
Only Path	Detour	Meters
Only Path	Normalized detour	Ratio
All Trajectories	Time of day	Seconds since midnight

Table 5.1: Examples of features

5.1.3 Models

The implemented models determine the probability of a feature value based on the set of trained trajectories with the same context. These features can be the speed, orientation, length or any other property that can be extrapolated from a trajectory between two points. Examples of these features can be seen in Table 5.1.

5.1.4 Trajectory anomalies

The combination of a Gaussian model, personal detector and the speed as feature tries to find individuals who suddenly changed their speed due to an event. A persons speed could well be normally distributed since everybody has their own preferable walking speed but crowded areas where the speed is lower or points of interest, such as a street full of shops can cause outliers according to this model.

For the shopping street scenario, an edge detector modelling speed as Gaussian distributed would be a good alternative to detect anomalies in the street. The edge detector creates a model of everybody in that particular street. People running in a street where the usual walking speed is much lower will be outliers and marked as anomalies.

The ability to combine all feature based anomaly detectors with all implemented models and features opens the possibility to detect anomalies that are not thought

of before.

5.2 Collective detectors

Apart from looking at individual trajectory features, a collective model represents the location of the objects and their relation to each other at one moment in time. The densities at a number of successive time frames are compared in a moving window to determine what locations show collective anomalous behaviour.

5.2.1 Input

The input for collective detectors are frames instead of trajectories. Frames are measurements received at or about the same time thus represents the locations of people at one moment in time.

5.2.2 Detectors

The detectors determine the collective representation and context. For example a density detector will, as the name suggests, compare the amount of people in different areas. For this detector an area will be anomalous or not, not an individual person. Neighbourhood or dissimilarity detectors are personal and not location based, they evaluate for example what their distance to the k people closest to them is and either compare this to other people's *neighbourhoods* or take the change in the personal neighbourhoods over time.

5.2.3 Windows

Time windows are used for evaluating changes over time, either by determining the maximal value (density, neighbourhood boundary distance, etc.), the increase or decrease or even frequency if desired. Anomalies are detected when the outcome of a window passes a given threshold.

5.2.4 Collective anomalies

Using this approach, it is to find locations where within a certain amount of time (decided by the window size), the density changes due to people coming together or running away. Last mentioned scenario could occur when people witness somebody caring a weapon or bomb.

Chapter 6

Results

This chapter will cover the performed tests and their results. For both statistical and collective anomaly detectors, two types of tests were performed. One to analyse how well the models are able to capture the normal data and another to check if simulated events are detectable.

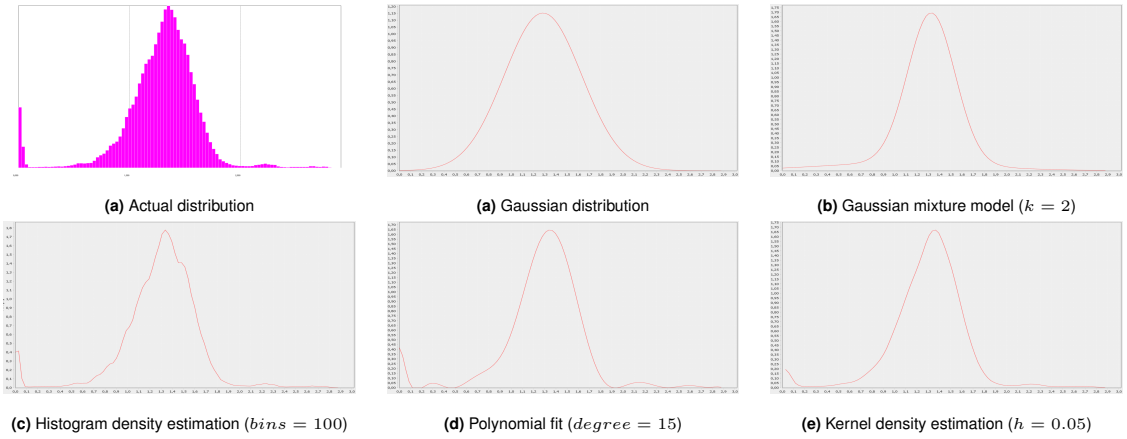
6.1 Normal models

The 4 hour simulation without any generated events has in total 952609 segments for 18230 agents (fictional people). Since this simulation only contained normal agents and no events were present, the less false positive anomalies detected, the better the model fits the normal data.

6.1.1 Statistical models

We can inspect how well the models capture the data visually. In Figure 6.1 we can see the distribution of the global speed and all probability density functions generated by the same data.

Apart from minor differences, all probability density models (Figure 6.1b-e) fit the actual distribution (Figure 6.1a) well. This results in near zero anomalies detected for the models, based on global detection, as can be seen in Figure 6.2. Detectors based on different context show less well fitting models up to about 35% false pos-



(a) Actual distribution **(a)** Gaussian distribution **(b)** Gaussian mixture model ($k = 2$)

(c) Histogram density estimation ($bins = 100$) **(d)** Polynomial fit ($degree = 15$) **(e)** Kernel density estimation ($h = 0.05$)

Figure 6.1: The different distribution models of all speeds over 4h.

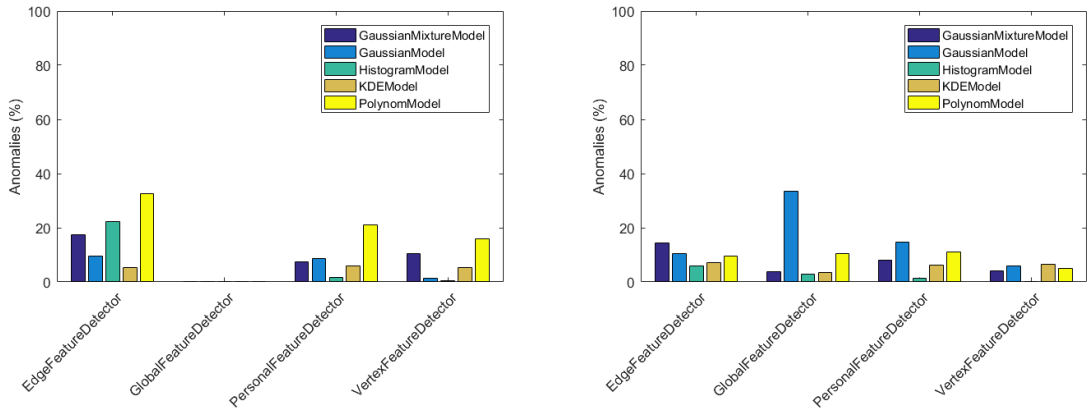


Figure 6.2: Percentage of the anomalies found in normal data for features speed (left) and direction (right), with for all a threshold of 0.01

itives for a Gaussian distribution for global detection of the direction of trajectories. This result is not surprising because it would be very unlikely to have every person in a city walking only in one direction.

The peak close to zero is caused by people standing still. They are excluded from the test set for Figure 6.2 to only test the moving trajectories. Detection based on people standing still can both be done by trajectory or collective detection where another feature such as location is used to look at whether people stand still on unusual places.

Figures 6.3 & 6.4 show the same false positives as Figure 6.2, but now measured over time. This shows how some combinations are adjusting to the new data and improve their statistical model according to the data where others are getting worse because the data distributed ideally for the model.

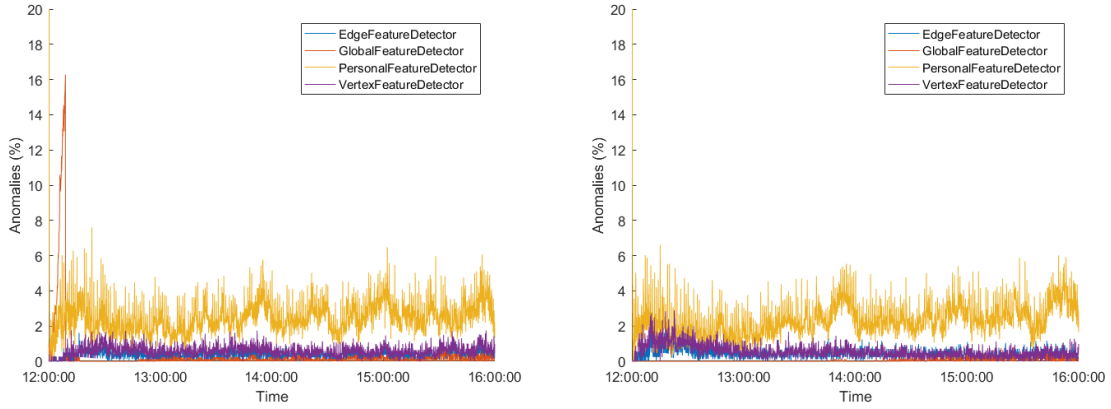


Figure 6.3: Percentage of the anomalies found in normal data based on histogram models over time. Left is based on feature speed and right on direction, with for all a threshold of 0.01

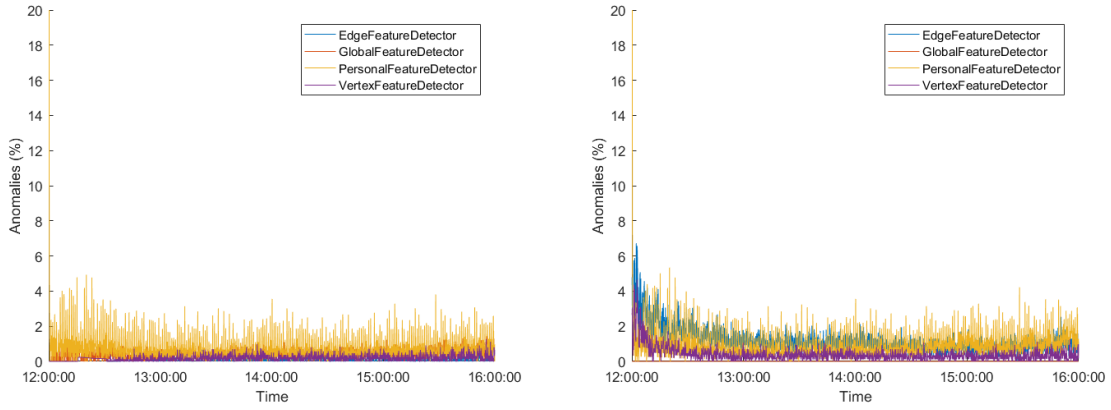


Figure 6.4: Percentage of the anomalies found in normal data based on Gaussian models over time. Left is based on feature speed and right on direction, with for all a threshold of 0.01

6.1.2 Collective models

The collective models require different testing. Instead of comparing trajectory features, this technique uses the locations of every person with each new measurement[1]. Therefore the tests are done in chronological ordering, as if the data is acquired and analysed in real time.

The first test will do this by adding the frames in series. After a frame has been added the models are updated and the anomalies are calculated for 4 different thresholds. What percentage of the scene will be marked as anomaly can be seen in Figure 6.5.

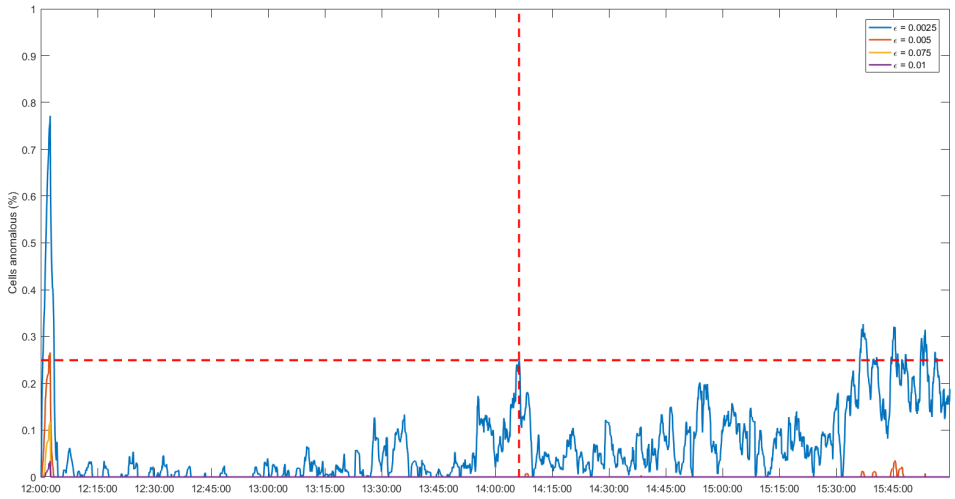


Figure 6.5: The anomalies found based on three different thresholds. All use a window size of 30 frames. The vertical line represents the snapshot visible in Figure 6.6.

To give a more detailed view in what is tested, one frame taken during this test is shown in Figure 6.6. Left is the actual position of the simulated agents and right the areas where the densities changed more than 0.0025 in the last 30 frames. By comparing this with the dotted line in Figure 6.5, we can see the anomalous cells currently covers about 0.25% of the whole area.

The relatively high anomaly peek in the first minute of Figure 6.5 is again due to the start-up of the test. This case the reason is not the lack of data to generate a

[1] Technically, the combination of time and location could be expressed as context and feature of a trajectory as well. More research on whether the trajectory detection technique is able to replace the collective detection technique can be done (see Chapter 9)

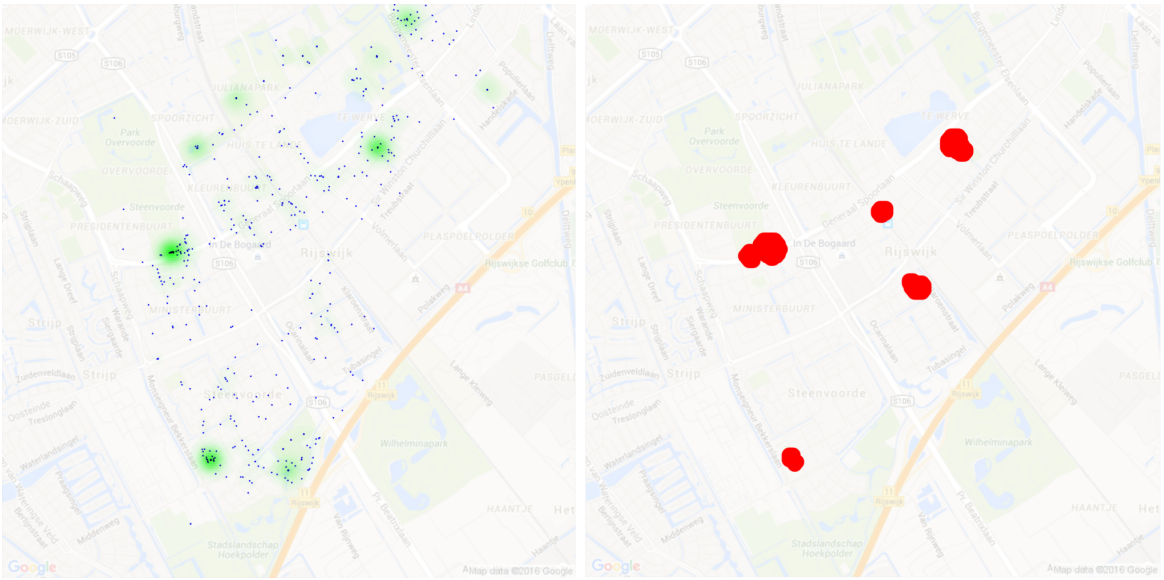


Figure 6.6: Left: The densities of one frame using KDE. The greener the area, the denser it is. The blue dots indicate the location of people in this frame. **Right:**The anomalies found using a 30-frames window and a threshold of 0.0025. The red areas are the locations where the density changed more than 0.0025 in the last 30 frames.

reliable model but the simulated data has everybody starting at a limited number of places at the same time. Therefore these peaks are not false positives but actual anomalies generated by this test setup. After the people are doing what their agenda tells them to do the number of anomalies decreases.

6.2 Simulated events

6.2.1 Procession

Detecting of a procession would therefore be expected to work best using a collective anomaly detection approach. Especially when a dense group of people is walking, the density in an area changes significantly. Spikes in Figure 6.7 show the increase in anomalous areas at the times of the events. Whether the spikes are actually detecting the correct location is measured by averaging the distances between the anomalous locations and the starting location at the time of the event.

Figure 6.8 & Table6.1 show how a higher threshold will mark a smaller area close

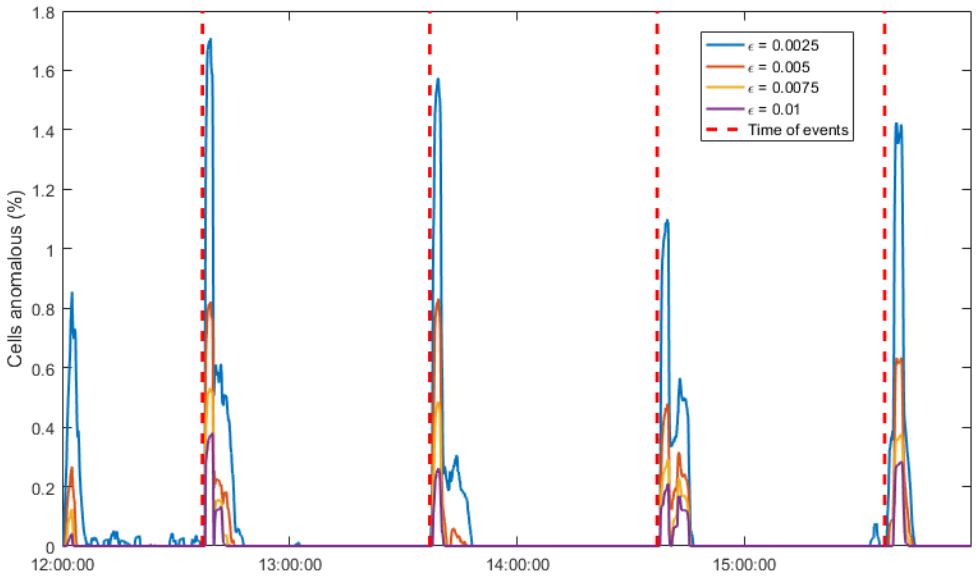


Figure 6.7: The anomalies found based on four different thresholds. All use a window size of 30 frames.

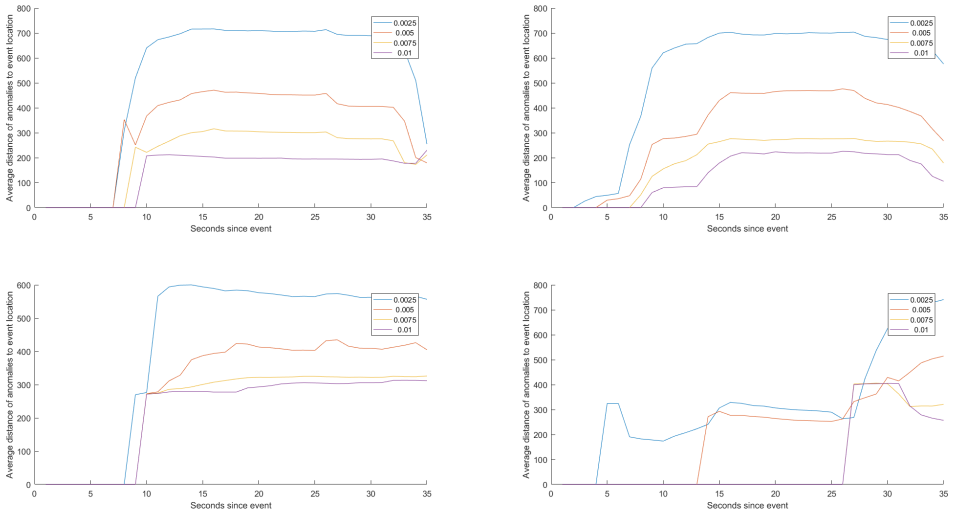


Figure 6.8: The distances to the event of anomalies found based on four different thresholds. All use a window size of 30 frames.

to the origin of the event as anomalous, but will take more time to detect the anomaly.

Threshold	12:37	13:37	14:37	15:37	Score
$\epsilon = 0.0025$	652,69	573,56	552,00	363,47	504,95
$\epsilon = 0.005$	405,52	355,23	392,41	332,18	364,37
$\epsilon = 0.0075$	275,32	239,96	313,35	360,16	315,40
$\epsilon = 0.01$	198,67	177,31	295,12	348,05	274,02

Table 6.1: The average distances from all locations marked as anomaly to the starting location of the event for the 30 frames after the event starts.

6.2.2 Pickpocketing, robbery and street theft

The event repeats 4 times with each 2 offenders. The maximal number of successful detection for the detectors is therefore 8 agents.

The plots in Figure 6.9 and colour map in Figure 6.10 show how the different detectors and models respond to the 4 theft events. The most successful detectors will not mark the offenders as anomalies before the event takes place (at 0) and detect all 8 shortly after. How long they keep marking the agents as anomalies depends on when the offenders 'blend in' or go inside.

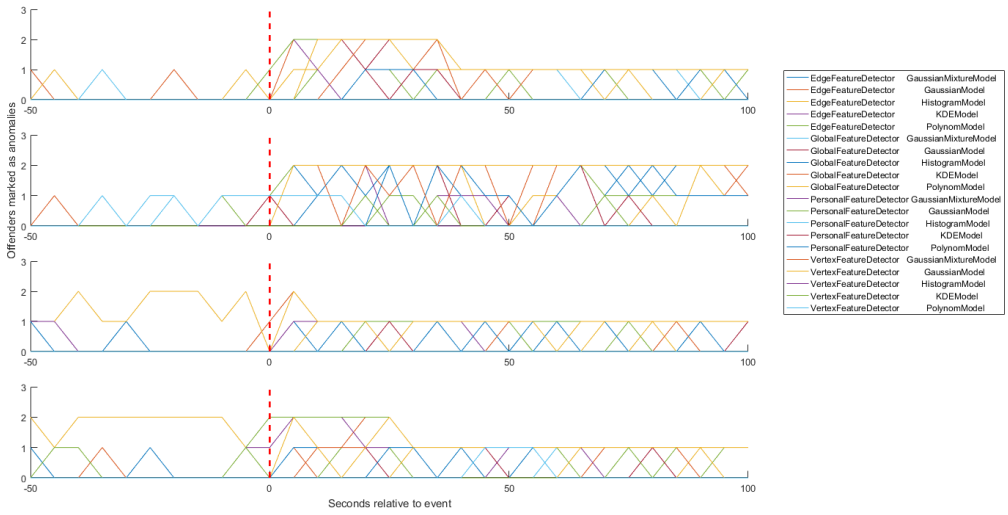


Figure 6.9: The anomalies found in the different events for 4 detectors and 5 methods, based on the feature speed.

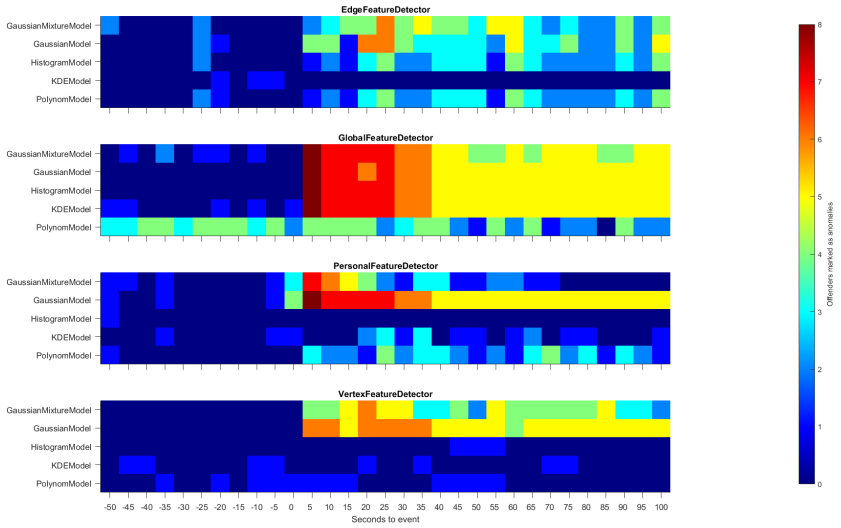


Figure 6.10: The accumulated anomalies found for 4 detectors and 5 methods, based on the feature speed, shown as colour map.

6.2.3 Shoplifting and commerical robbery

As for pickpocketing, the test event repeats 4 times with each 2 randomly selected offenders. The maximal number of successful detection for the detectors is therefore 8 agents. Figure 6.11 shows when the offenders are detected as anomalies for the four different events. In Figure 6.12, these events are combined to show the total detected offenders for the different combination of detector and model.

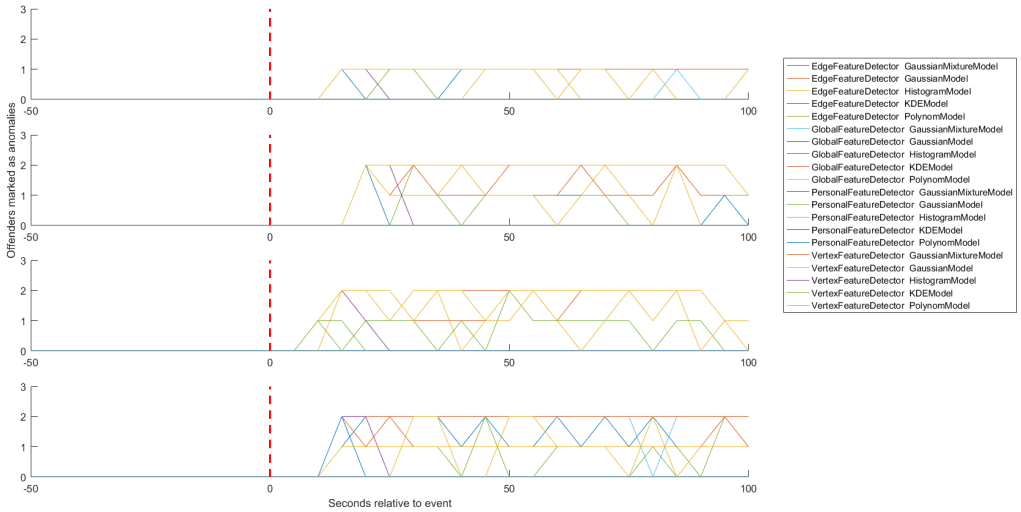


Figure 6.11: The anomalies found in the different events for 4 detectors and 5 methods, based on the feature speed.

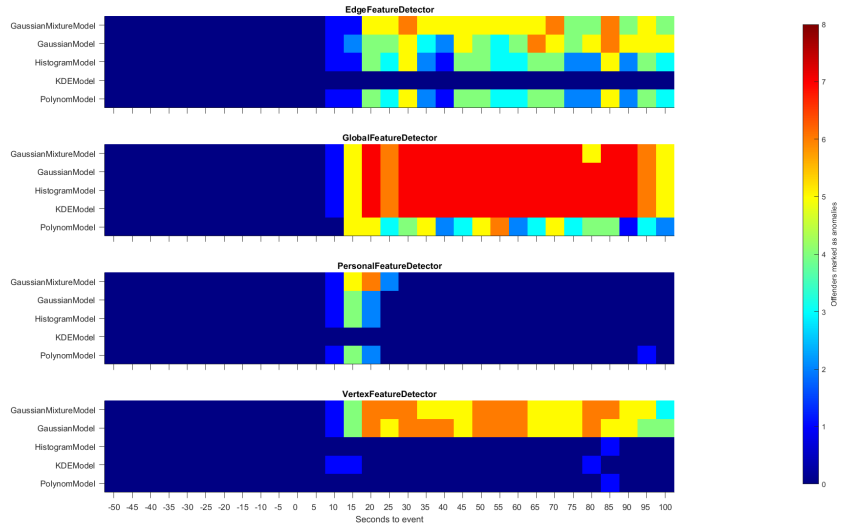


Figure 6.12: The accumulated anomalies found for 4 detectors and 5 methods, based on the feature speed, shown as colour map.

Chapter 7

Discussion

7.1 Model sizes

The comparison of the combinations between detector and method (Figure 6.2) shows big difference in success. False positives are rarely seen in the global detector but high in for example the personal detector. One important reason for the high number of anomalies in personal detection is the limited amount of data. For every person, the first trajectory is an anomaly because no model for this person exists. This is even worse for the models that require multiple trajectories with different feature values such as the Polynomial model. Table 7.1 shows this the combination of number of models and trajectories per model for the different detectors based on the same data set used for testing. The higher the number of models and the lower the number of trajectories per model, the higher likeliness of false positives.

An important question that has to be answered based on these findings is what to do with these 'empty models'. A solution for this problem is to use models only when they contain enough data to be considered reliable. Until then, the global detector

Detector	Models	Average	Standard Deviation
Global Detector	1	952609	0
Personal Detector	18230	52.26	168.79
Vertex Detector	9888	96.34	181.16
Edge Detector (if directed)	34156	27.89	65.39
Edge Detector (if undirected)	24511	38.86	85.05

Table 7.1: The amount of models per detector and trajectories per model in the data set.

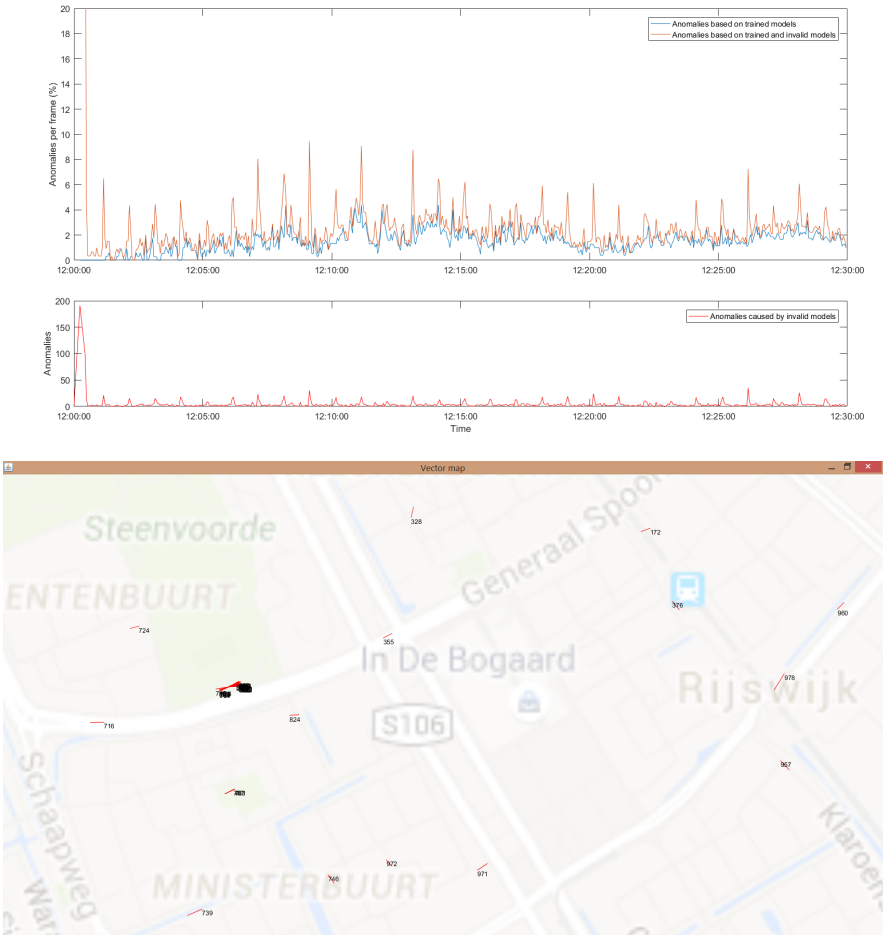


Figure 7.1: Anomalies found using a personal detector with Gaussian models. The map shows how the high peeks in both graphs are caused by newly detected agents.

can work as an initial model. Either by switching from global to personal detection when the models contain sufficient amount of data or by initializing the personal model with the same parameters as a global model and re-estimating parameters when new data is added to the personal model.

Figure 7.1 shows the difference between considering trajectories of newly detected people an anomaly compared to ignoring anomalies due to invalid (empty) models. The amount of false positives caused by invalid models shows a repetitive pattern. Every minute it has a spike, suggesting the simulator adds the new agents every minute and not evenly distributed across time.

7.2 Order of training and detection

One of the solutions for this problem is to add any newly received data to the model before checking whether it is an anomaly. This will increase the validity of the model but will also have unwanted false negatives since the exact value you are testing is used in training. There are actually several options on what order training and testing are done, as well as whether an anomaly should be added to the model or not. In general we have to ask the question if and when should an anomaly be added to the model? The options are:

1. First add a trajectory to the model, then check if it is an outlier.
2. First add a trajectory to the model, then check if it is an outlier. If it is an anomaly, remove it from a model.
3. First add a trajectory to the model, then check if it is an outlier and ask feedback from the user whether this is an anomaly. Subsequently remove it if it is an anomaly according to the domain specialist.
4. First detect if the trajectory is an outlier, subsequently add it to the model.
5. First detect if the trajectory is an outlier, then leave it out of the model.
6. First detect if the trajectory is an outlier, ask feedback from the domain specialist whether this is an anomaly and if it is, leave it out of the model.

Ideally, the domain specialist would select whether the outlier is an anomaly or not but not all situations would have this possibility. The cases where lots of outliers are detected would benefit but will require a lot of error checking by the operator. Some situations would also not immediately give a clear answer on whether it is an anomaly or not. It could require eyes on the ground and review of the outlier, which would only be possible when there are a small number of outliers.

The options where outliers are not used would make the model most accurate but will be hard to accomplish. For example, in the beginning the models will mark every first trajectory as outlier because it has no data yet. Since it is an outlier it will not be added to the model and the same problem will arise for the next trajectory. For these situations it would make more sense to first add them until a stable model is created and subsequently remove the outliers.

7.3 Simulator

After some events, agents kept standing still at the event location. However, it is more likely to have offenders and/or victims run in if these situations occur in real life. Whether this behaviour is indeed what happens is open for discussion. Other testing parameters such as the locations of the generated events are picked at random. Whether these locations are plausible for the events and how much it matters what location the event has, is debatable. Some events, such as a demonstration, more likely start at an important square compared to a small alley somewhere in the area. During testing these locations are not prioritized and all events have a unique location to make the tests independent. For example, the collective densities will already be high before the second test starts if they occur at the same location, a relatively small increase caused by the event will not be detected.

Since several tests are done based on the speed of the agents, their speed should represent a realistic representation of real live situations. Modifications to the simulator were required to accomplish this (see Appendix B), but whether the resulting speeds are in fact a likely representation will remain questionable.

As mentioned earlier, agents are spawning at a fixed rate, namely every minute. Furthermore they only spawn at a limited number of places when the simulator is started. later they do have a better and more widely spread distribution.

Chapter 8

Conclusions

8.1 Outliers vs anomalies

The characteristics of crimes are often hard to define. Not every person running out of a bank committed a robbery, instead he might be running to catch the train. Running out of a bank has therefore no direct relation to robbing the bank. On the other site, if other people (for example police officers) were witnessing the situation, they might want to question the person why he runs out of a bank. This behaviour is in this case perceived as abnormal or suspicious. Other suspicious looking behaviour includes suddenly running away from somebody and this person can be perceived as offender (of a street theft) or potential victim of something the other person did.

Research on what other behaviour people and specifically domain specialist such as police officers or security camera operators perceive as suspicious is needed to find more characteristics of anomalous behaviour. The two examples mentioned above are, as well as characteristics of a procession were used during this research. They are defined as follows:

Street robbery The offenders walk towards the victims and start running after the offence.

Commercial robbery The offenders walk inside a store and run away after the offence.

Procession The participants come together at one location and start walking the same route.

Examples of the events mentioned in Chapter 2 and how they might be detected are listed in Tables 8.1 & 8.2. Some events could be detected using different combinations of detectors, features and models. Some of these techniques will also detect anomalous events that are uncommon but are not a crime. For example, a personal detector checking for the detour somebody took to get a bicycle, can detect if a thief is looking for a one to steel but will also detect if somebody simply forgot where his bicycle was parked.

The tables also show detectors not yet fully realized in the current prototype. See future work (Section 9 and Section A.5) for more details on what these detectors should do.

Detector	Feature	Example of detectable events
Global	Speed	Running
Edge	Speed	Shoplifting and commercial robbery
Edge	Direction	Walking against traffic
Edge	Direction	Pickpocketing
Personal	Speed	Street theft
Vertex	Speed	Stalking
Vertex	End-location	Burglary
Personal	Detour	Bicycle theft

Table 8.1: Examples of anomaly types using trajectories

Detector	window	Anomaly type
Collective density	Density increase	Rally or demonstration
Collective density	Density decrease	Terrorist spotted
Collective dissimilarity	Repetition	Pickpocketing
Collective dissimilarity	Distance + Variance	Stalking

Table 8.2: Examples of collective anomaly types

8.2 Detection technique

The techniques used in this research for detecting anomalies has proven to work for both trajectories and collective behaviour. The different anomaly characteristics can be translated into a context-model-feature combination which can detect a certain specific anomaly type.

Context based grouping of trajectories (which will create one model for each group) works well, as long as the groups contain enough training data for a valid and reliable model. Some models require more data and therefore more start-up

time than others but not all contexts can be properly trained by extending the start-up time. Some contexts such as personal context (or anything that is related to time) have a short lifespan, making the model unreliable for a significant part of the time the context exists. The different ways features are distributed requires the possibility to choose from different models. A Gaussian model works well for a feature like speed but will not be able to model the distribution of the directions, for example.

Collective detection is a slightly different technique where spacial models are constructed at a fixed frame rate. This method has proven to work well when models are compared over time to detect changes like the density (crowdedness) of people.

8.3 Simulatable events

Currently, the simulator was able to provide three types of events interesting for anomaly detection: A procession, street robbery and a commercial robbery. The simulator is able to perform more events, such as a police arrest but for anomaly detection as tool to find criminal activity an arrest is less useful since in this case, the police already found the offender.

To answer the question on how well the methods mentioned in this research are able to find the generated anomalous events, we will look at those individually and compare the different detectors.

8.3.1 Street robbery using trajectory detection

The distinct characteristics of this event are well detectable for global analysis of all trajectories. The speed of the fleeing offenders is considered an outlier for all models except the polynomial fit function.

Detectors based on personal context work best with the Gaussian model and GMM. The KDE and polynomial model do show an increased detection rate after the event but only half as strong as the Gaussian model. Where the histogram did work well without context (Global detection), it seems to fail for personal detection. GMM shows successful detection but the offenders will not be marked as anomalies as long as they are in a single Gaussian model. This is caused by the adaptation of the models, they consider both the walking and running behaviour as normal.

Edge detectors do not capture this event well. A small increase in successful detection is visible but less profound as the global and personal detectors do. KDE is the exception in this situation since it did not capture any anomaly during the events.

Vertex detection performs better than edge detection, most likely because the models are better trained because more people crossed these vertices compared to the same edges as the offender did.

8.3.2 Commercial robbery using trajectory detection

Global detection based on speed is for this event also the method of choice. Compared to the street robbery we can notice some small differences but these are caused by the randomness of the simulator, the people picked as offenders and their final destination.

As expected does the personal detector a bad job in detecting the events. The offender has an 'empty model' (no information about what normal behaviour is for this person), therefore the running speed is considered normal quickly after exiting the building.

Although not as good as for the global models do the Gaussian an GMM work in vertex detection as well. The other models seem to work bad for the test simulation. This is most likely caused by insufficient data for the vertices the offenders pass. Not enough people passed these hubs to generate a reliable model. On the other hand, an edge connects two vertices but the edge detector does perform better for these methods.

One of the first noticeable things is the lack of false positives before the event takes place. The actual reason for this is the loss of tracking during the event. In other words, there is no information about the offender before the event. The detection of the anomalous person is also slightly later compared to the street robbery. This is caused by the time it takes for the offender to go outside after the offense but a delay could also be caused by empty or still invalid models, for example in personal detection.

8.3.3 Procession using collective detection

Detecting the procession using collective detection methods works well. All 4 thresholds show anomalies just after the events (Figure 6.7) but the delay is longer for a higher threshold (Figure 6.8). On the other hand, the detection of the higher thresholds is more accurate (Table 6.1). Deciding what threshold is optimal depends on the priority of the operator, whether he prefers speed above accuracy or not.

Chapter 9

Recommendations and future work

To be able to detect different crimes, distinguishing characteristics of those crimes need to be found. For example, in this research the assumption is made that offenders will flee (run) after they committed the crime. However, as explained in Chapter 2, this is not always the case. Research on what behavioural patterns are typical for the types of crimes will be required to be sure you are simulating and detecting criminal behaviour. Based on the characteristics, the context, feature and model can be chosen and proven to detect the crime related to those characteristics.

In this research, the focus was on detecting anomalies in wide areas, but different parts of this area will require other priorities what anomalies should be detected. Small use cases are recommended for the individual crimes to better find the correct parameters and threshold values. Subsequently they can be combined to detect anomalies on a larger scale.

The implemented methods were tested using simulated data and therefore we can not assume the simulation is fully correct. Conclusions on whether the presented methods would also work in real life situations can only be tested with actual tracking data. This would require sufficient background checks and cause other complications due to the classified data. For this reason the simulation was the best alternative test method. Another way to collect data without the simulator is by acting out the different situations (in a small controlled environment). Use cases for the individual events will have to be predefined based on the characteristics typical for the crime as well. Real life tracking situations will also entail false input, for example wrongly detected objects or missing data. This is not included or simulated in this research but can cause an anomaly detector to classify incorrectly.

In those situations where people can be labelled as anomaly, either because they are simulated or acting, the method used during this research can be used in a neural network for example. The inputs for the first layer would be the set of trajectories, context, model (including parameters) and feature and the outcome would be a list on anomalous trajectories. The rest of the layers could be a detector again, with another combination of context, model and feature or be like the ensemble from Figure 3.3 and Figure 5.1.

Throughout the research, assumptions were made of what the anomalies are. Whether these are actually anomalies a domain expert would be interested in was not taken into account. Research on what the anomalies are and which of them are not relevant for an operator or domain expert, would be recommended.

Although this research did include anomaly detection methods used in other domains, another detailed analysis of what context, models and features work well for those situation can benefit research in anomaly detection for defence and surveillance purposes.

The time necessary to implement all mentioned techniques was unfortunately longer than the available time for the research. The design of the current detectors can be used as framework and enables the implementation of alternative methods both for variations on current implementations such as other statistical methods as for new areas like using multiple dimensional statistical methods to find correlations.

Currently a distinction is made between trajectory and collective detection but they are in fact more similar than it looks. The collective detection method creates a model for every frame, which is subsequently compared to earlier frames. If those time frames are seen as the contextual property of a trajectory and the end position of the trajectory as feature, the trajectory detection method would be able to detect the same anomalies. However, two aspects of the collective detection are not yet possible in trajectory detection which is why this research makes a distinction between the two techniques. The trajectory detection should be able to compare models of different contexts to detect the changes over time and the two-dimensional aspect of collective detection has to be translated into a multivariate feature distribution where currently only one-dimensional PDFs were used.

Another recommended new detector would be one that uses graph and shortest path theory to detect whether a person did not take the optimal path. This detector will most likely not be able to work with the current statistical models or basic feature extraction. Either the model will need to keep track of and be able to calculate the

fastest route or the detour has to be calculated during feature extraction. Other suggestions and already partly realized detectors are explained in Appendix A.

Bibliography

[1] B. Leininger, “A next-generation system enables persistent surveillance of wide areas,” *SPIE Newsroom*, 2008. [Online]. Available: http://www.spie.org/x23645.xml

[2] GTI, “Global Terrorism Index 2015: Measuring and understanding the impact of terrorism,” 2015.

[3] G. K. Still, “Crowd dynamics,” Ph.D. dissertation, University of Warwick, 2000. [Online]. Available: http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.364679

[4] K. M. Monk, J. A. Heinonen, and J. E. Eck, *Street robbery*. Washington, DC: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2010, oCLC: 747034971.

[5] R. V. Clarke and G. Petrossian, *Shoplifting*, 2nd ed. Washington, D.C.: U.S. Dept. of Justice, Office of Community Oriented Policing Services, Apr. 2013. [Online]. Available: www.cops.usdoj.gov

[6] D. L. Weisel, *Bank robbery*. Washington, DC: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2007, oCLC: 318929155.

[7] T. Keister, *Thefts of and from cars on residential streets and driveways*. Washington, DC: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2007, oCLC: 809753810.

[8] D. L. Weisel, *Burglary of single-family houses*. Washington, D.C.: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2004, oCLC: 76818423. [Online]. Available: http://purl.access.gpo.gov/GPO/LPS76311

[9] S. D. Johnson, A. Sidebottom, and A. Thorpe, *Bicycle theft*. Washington, DC: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2008, oCLC: 277000523. [Online]. Available: http://www.popcenter.org/problems/pdfs/bicycle_theft.pdf

[10] S. Herman and M. Markon, *Stalking*. Washington, D.C.: U.S. Dept. of Justice, Office of Community Oriented Policing Services, 2004, oCLC: 61314779.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," 2007.

[12] H. Teng, K. Chen, and S. Lu, "Adaptive real-time anomaly detection using inductively generated sequential patterns," in *, 1990 IEEE Computer Society Symposium on Research in Security and Privacy, 1990. Proceedings*, May 1990, pp. 278–284.

[13] J. Janssens, "Outlier selection and one-class classification," Ph.D. dissertation, Tilburg University, Tilburg, 2013. [Online]. Available: http://jeroenjanssens.com/jeroenjanssens-thesis.pdf

[14] J. B. Kraiman, S. L. Arouh, and M. L. Webb, "Automated anomaly detection processor," vol. 4716, 2002, pp. 128–137. [Online]. Available: http://dx.doi.org/10.1117/12.474940

[15] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2347755

[16] B. J. Rhodes, N. A. Bomberger, M. Seibert, and A. M. Waxman, "Maritime situation monitoring and awareness using learning mechanisms," in *IEEE Military Communications Conference, 2005. MILCOM 2005*, Oct. 2005, pp. 646–652 Vol. 1.

[17] O. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition #x2014;A Review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 865–878, Nov. 2012.

[18] Z. He, S. Deng, and X. Xu, "Outlier Detection Integrating Semantic Knowledge," in *Advances in Web-Age Information Management*, ser. Lecture Notes in Computer Science, X. Meng, J. Su, and Y. Wang, Eds. Springer Berlin Heidelberg, Aug. 2002, no. 2419, pp. 126–131, dOI: 10.1007/3-540-45703-8_12. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-45703-8_12

[19] Z. He, J. Z. Huang, X. Xu, and S. Deng, "Mining Class Outliers: Concepts, Algorithms and Applications," in *Advances in Web-Age Information Management*, ser. Lecture Notes in Computer Science, Q. Li, G. Wang, and L. Feng, Eds. Springer Berlin Heidelberg, Jul. 2004, no. 3129, pp. 589–599, dOI: 10.1007/978-3-540-27772-9_59. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-27772-9_59

[20] M. K. Saad and N. M. Hewahi, “A comparative study of outlier mining and class outlier mining,” *CS Letters*, 2009.

[21] R. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[22] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition,” *Image and Vision Computing*, vol. 14, no. 8, pp. 609–615, Aug. 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0262885696011018

[23] B. A. Olshausen and D. J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996. [Online]. Available: http://www.nature.com/doifinder/10.1038/381607a0

[24] L. Parra, G. Deco, and S. Miesbach, “Statistical Independence and Novelty Detection with Information Preserving Nonlinear Maps,” *Neural Computation*, vol. 8, no. 2, pp. 260–269, Feb. 1996. [Online]. Available: http://dx.doi.org/10.1162/neco.1996.8.2.260

[25] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[26] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[27] J. Verbeek, “Mixture models for clustering and dimension reduction,” phdthesis, Universiteit van Amsterdam, Dec. 2004. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00321484/document

[28] L. Kratz and K. Nishino, “Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models,” in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, Jun. 2009, pp. 1446–1453.

[29] L. J. Latecki, A. Lazarevic, and D. Pokrajac, “Outlier Detection with Kernel Density Functions,” in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner, Ed. Springer Berlin Heidelberg, Jul. 2007, no. 4571, pp. 61–75, dOI: 10.1007/978-3-540-73499-4_6. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-73499-4_6

[30] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artif Intell Rev*, vol. 22, no. 2, pp. 85–126, Oct. 2004. [Online]. Available: http://link.springer.com/article/10.1007/s10462-004-4304-y

[31] B. Morris and M. Trivedi, "A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008.

[32] Z. Ferdousi and A. Maeda, "Anomaly Detection Using Unsupervised Profiling Method in Time Series Data." in *ADBIS Research Communications*. Citeseer, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.4348&rep=rep1&type=pdf

[33] B. G. Lindsay, "Mixture Models: Theory, Geometry and Applications," *NSF-CBMS Regional Conference Series in Probability and Statistics*, vol. 5, pp. i–163, 1995. [Online]. Available: http://www.jstor.org/stable/4153184

[34] R. Laxhammar, "Anomaly detection for sea surveillance," in *2008 11th International Conference on Information Fusion*, Jun. 2008, pp. 1–8.

[35] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional Anomaly Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 5, pp. 631–645, May 2007.

[36] F. Jiang, Y. Wu, and A. K. Katsaggelos, "Detecting contextual anomalies of crowd motion in surveillance video," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Nov. 2009, pp. 1117–1120.

[37] M. Markou and S. Singh, "Novelty detection: a reviewpart 2:: neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, Dec. 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168403002032

[38] W. K. Hrdle, M. Mller, S. Sperlich, and A. Werwatz, *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.

[39] D. M. J. Tax, *One-class classification: concept-learning in the absence of counter-examples*. [S.l.]: [s.n.], 2001. [Online]. Available: http://homepage.tudelft.nl/n9d04/thesis.pdf

[40] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004. [Online]. Available: http://www.springerlink.com/index/jl3n42j6067703n2.pdf

[41] J. H. M. Janssens, F. Huszr, E. O. Postma, and H. J. van den Herik, "Stochastic outlier selection," Technical report TiCC TR 2012-001, Tilburg University, Tilburg Center for Cognition and Communication, Tilburg, The Netherlands, Tech. Rep., 2012. [Online]. Available: https://www.tilburguniversity.edu/upload/b7bac5b2-9b00-402a-9261-7849aa019fbb_sostr.pdf

[42] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-based Local Outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/342009.335388

[43] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques*, 2nd ed., ser. Morgan Kaufmann series in data management systems. Amsterdam ; Boston, MA: Morgan Kaufman, 2005.

[44] R. Llet, M. C. Ortiz, L. A. Sarabia, and M. S. Snchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," *Analytica Chimica Acta*, vol. 515, no. 1, pp. 87–100, Jul. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003267003016246

[45] G. Chen, S. A. Jaradat, N. Banerjee, T. S. Tanaka, M. S. Ko, and M. Q. Zhang, "Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data," *Statistica Sinica*, pp. 241–262, 2002. [Online]. Available: http://www.jstor.org/stable/24307044

[46] C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, Nov. 1995.

[47] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "iBAT: Detecting Anomalous Taxi Trajectories from GPS Traces," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, ser. UbiComp '11. New York, NY, USA: ACM, 2011, pp. 99–108. [Online]. Available: http://doi.acm.org/10.1145/2030112.2030127

[48] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[49] D. E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning*," *Cognitive Science*, vol. 9, no. 1, pp. 75–112, Jan. 1985. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0901_5/abstract

[50] J. Hawkins, “Hierarchical temporal memory including HTM cortical learning algorithms,” *Techical report, Numenta, Inc*, 2011. [Online]. Available: http://numenta.com/learn/hierarchical-temporal-memory-white-paper.html

[51] A. Basharat, A. Gritai, and M. Shah, “Learning object motion patterns for anomaly detection and improved object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, Jun. 2008, pp. 1–8.

[52] M. Anneken, Y. Fischer, and J. Beyerer, “Evaluation and comparison of anomaly detection algorithms in annotated datasets from the maritime domain,” in *SAI Intelligent Systems Conference (IntelliSys), 2015*, Nov. 2015, pp. 169–178.

[53] C. Brax, “Anomaly Detection in the Surveillance Domain,” Ph.D. dissertation, University of Skovde, 2011. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:431243/FULLTEXT01.pdf

[54] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, “A system for learning statistical motion patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[55] X. Wang, K. Tieu, and E. Grimson, “Learning Semantic Scene Models by Trajectory Analysis,” in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, May 2006, no. 3953, pp. 110–123, dOI: 10.1007/11744078_9. [Online]. Available: http://link.springer.com/chapter/10.1007/11744078_9

[56] S. Wang and Z. Miao, “Anomaly detection in crowd scene,” in *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, Oct. 2010, pp. 1220–1223.

[57] Z. Fu, W. Hu, and T. Tan, “Similarity based vehicle trajectory clustering and anomaly detection,” in *IEEE International Conference on Image Processing, 2005. ICIP 2005*, vol. 2, Sep. 2005, pp. II–602–5.

[58] B. Ristic, B. L. Scala, M. Morelande, and N. Gordon, “Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction,” in *2008 11th International Conference on Information Fusion*, Jun. 2008, pp. 1–7.

[59] N. A. Bomberger, B. J. Rhodes, M. Seibert, and A. M. Waxman, “Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness,” in *2006 9th International Conference on Information Fusion*, Jul. 2006, pp. 1–8.

[60] B. J. Rhodes, N. A. Bomberger, and M. Zandipour, “Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness,” in *2007 10th International Conference on Information Fusion*, Jul. 2007, pp. 1–8.

[61] H. Fanaee-T and J. Gama, “Event labeling combining ensemble detectors and background knowledge,” *Progress in Artificial Intelligence*, vol. 2, no. 2-3, pp. 113–127, Jun. 2014. [Online]. Available: http://link.springer.com/10.1007/s13748-013-0040-3

[62] E. Meijer, “Your Mouse is a Database,” *Queue*, vol. 10, no. 3, pp. 20:20–20:33, Mar. 2012. [Online]. Available: http://doi.acm.org/10.1145/2168796.2169076

Appendix A

Design & implementation

Research can conclude whether a method works for a specific task in anomaly detection, but sometimes more information on how the system works is required for reproducibility and understandability. Therefore this appendix will demonstrate some of the core design choices made during this research. Requirements such as near real time analysis are not only beneficial to keep the testing time as short as possible, it also creates the possibility to use (part of) the system for live anomaly detection. Examples of design choices made to increase detection speed can be found in graph search and allocation as well as GPU processing.

A.1 Data structure

The data received about the positions of people, is collected into a dataset. This dataset can be persistent, in which all previously received data is kept or buffered, which works as first in first out (FIFO) throwing away data after a while. The persistent data is usable for testing but will quickly take up memory, therefore the buffered dataset is more applicable when the detector has to run for a longer time.

The dataset keeps track of received data and turns it into the structure visible in Figure A.1. As explained in Chapter 5, two data points from the same person will create a segment and contiguous segments for a path. For both the received positional data (called a time-point) and the segment between the time-points, a graph representation is constructed. In this graph, a group of closely related time-points is generalized to a vertex and segments to edges of the graph. Finally, a

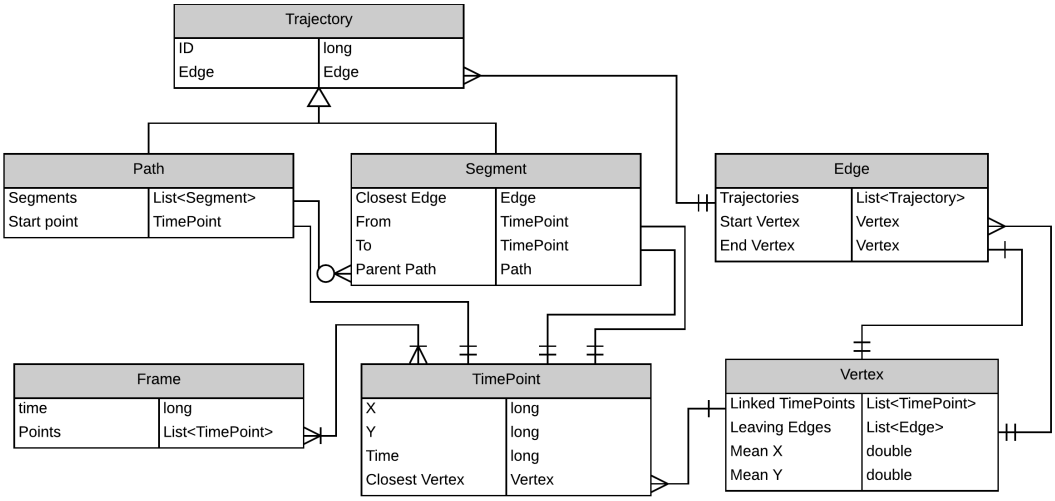


Figure A.1: Key aspects of the dataset model

Table A.1: The time it takes for a four hour simulation to load for different quad-tree parameters

		items per cell							
		1	2	5	10	15	20	25	30
Max depth of tree	5	169651	171367	171861	170985	172937	169437	170148	169204
	10	76953	79422	90727	89695	94232	102915	106555	111391
	15	77029	79056	82960	93877	98741	103706	105507	112402
	20	77372	79609	84156	91508	96153	104449	108446	111597

frame is the collection of time-points received at the same time.

A.2 Graphs

To create the graph, a quad-tree is used to find already existing vertices close to a newly received measurement. Whenever there are existing vertices close to the new data point, it is mapped to the one closest to the data point, otherwise a new vertex is added at this location. The graph has two important parameters: the maximal amount of points in a cell before it is split into four new cells, and the maximal depth of the tree. Picking the right parameters is crucial for optimal performance. The surface plot in Figure A.2 shows how the combination of the two parameters influence the time it takes to load the complete 4 hour test set. Keep in mind the maximum amount of items per cell is only kept as long as the particular cell is not at the maximum depth. Since every level of the tree is of degree 4, the amount of leave nodes is 4^d. If the points would be evenly spread in space, the quad tree would be able to hold $4^d \times max\_items$ points without breaking the maximal items per cell rule.

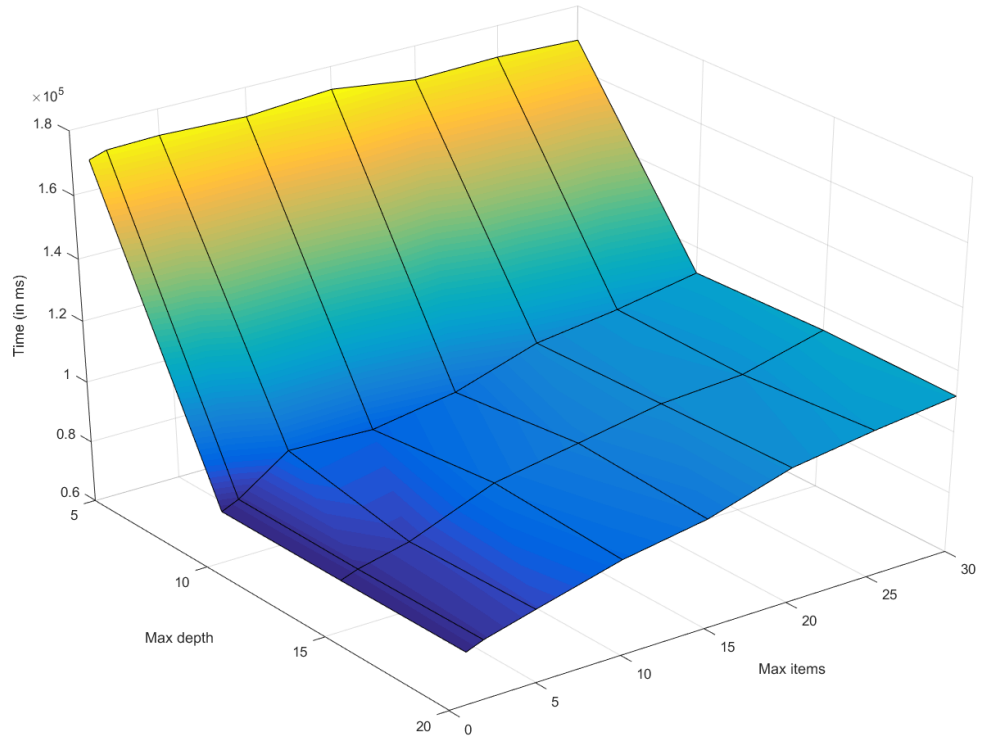


Figure A.2: The time it takes for a four hour simulation to load for different quad-tree parameters

By the high peek for low depth trees we can conclude that using a tree is preferred above having to go through all the items to find the nearest neighbours. On the other hand, having a low number of items per cell would cause a lot of overhead when cells have to be divided into four new cells. However, this splitting of cells is quicker for cells with less items compared to a high number of items per cell. Not taken into account here is the memory it takes for the different combination of parameters. The deeper the tree the more memory is required which should be kept in mind if memory is scarce.

A.3 GPU

Real time detection is key to the anomalies within the context of this research. If detectors are not able to calculate the anomalies in time, one of three options should be implemented in the design, based on what causes the delay:

- Data is added when available but the actual anomalies detection of the data is only done when computing power is available. Alerts of anomalies will be delayed by this method.

- Data is added when available but the models are updated once computing power is available. Anomaly detection is done based on old data, which can cause errors in detection.

- Data is only added when possible. Detectors are updated and detection is done subsequently. This will cause data loss and can both train invalid data of cause errors in detection.

To minimize the risk of delays caused by updating the detectors or detecting anomalies, GPU processing is used in the current design. Methods where independent calculations are required, such as calculating values for each cell in a grid, speeds up both updating and detection algorithms up to ten times.

A.4 Detector algorythms

A.4.1 Density based using KDE

One type of collective detectors uses a grid to determine a density for each cell in the grid individually. Currently the only implemented approach uses the principle of KDE. For each cell the euclidean distances to the different locations of people is calculated. Using a Gaussian kernel placed on top of this location, the densities are computed and added together. The result is a matrix with for each cell the density estimation.

Using the GPU to simultaneously compute the different rows of the grid optimizes this algorithm which makes the complexity for each row $grid_x \times N$ and in total $grid_x \times grid_y \times N$.

Windows to detect an increase or decrease in density compute the difference between the newest and oldest frame. A third window compares the minimal and maximal values for each grid cell over all frames and is able to detect differences in densities, regardless of whether this is an increase or decrease.

A.4.2 Neighborhood based using SOS

The second collective detector type calculates the relation between the points. Algorithms using this principle are for example K-NN or SOS. As explained in Chapter 3, SOS has a less strict boundary of exactly k neighbours, but rather has a perplexity h making points either more or less effective neighbours.

Stochastic Outlier Selection

The iterative process used in SOS computes the variances for each point to reach a fixed number of neighbours and the affinity and binding between them. After the initial dissimilarity matrix is calculated, the power of GPU processing can be used to find the affinity, binding probability, perplexity and variances for each point individually.

To reduce the number of iterations required to find the desired perplexity, a binary search is used. Initially the algorithm computes the perplexity for a variance which is in the middle of a given upper and lower boundary. If the computed perplexity is bigger than the required perplexity, the upper bound is set to the variance just used, otherwise the lower boundary is adjusted to the centre variance. After adjusting the boundaries the algorithm is repeated until either the perplexity is found or the maximum number of iterations is reached.

Computing the distance between points in an N-size dataset will require $N \times N$ calculations. The individual rows of the affinity matrix are computed simultaneously, but still require another N computations each. The same holds to normalize all values in the row, generating the binding matrix and again to calculate the perplexity of this point. An iteration of the algorithm after the dissimilarity matrix has been computed is therefore $3 \times N \times N$.

A.5 Recommendations

The current detectors are designed to cope with reading new positions as a live stream but, due to the possibility of evaluation and reproducibility, optimized for reading positions from file. This can be read at once or as a stream as well but real time detection is not an issue in these situations. Even though most methods were able to process (adding data to models and anomaly detection) in time, a highly suggested modification would be to implement streaming principles such as reactive programming[1] [62].

Recommended additions to the toolbox are primarily detectors for other context such as time as well as more types of collective detectors. See Figure A.3 for how these and other recommendations would relate to the already implemented methods and detectors.

[1] http://reactivex.io/

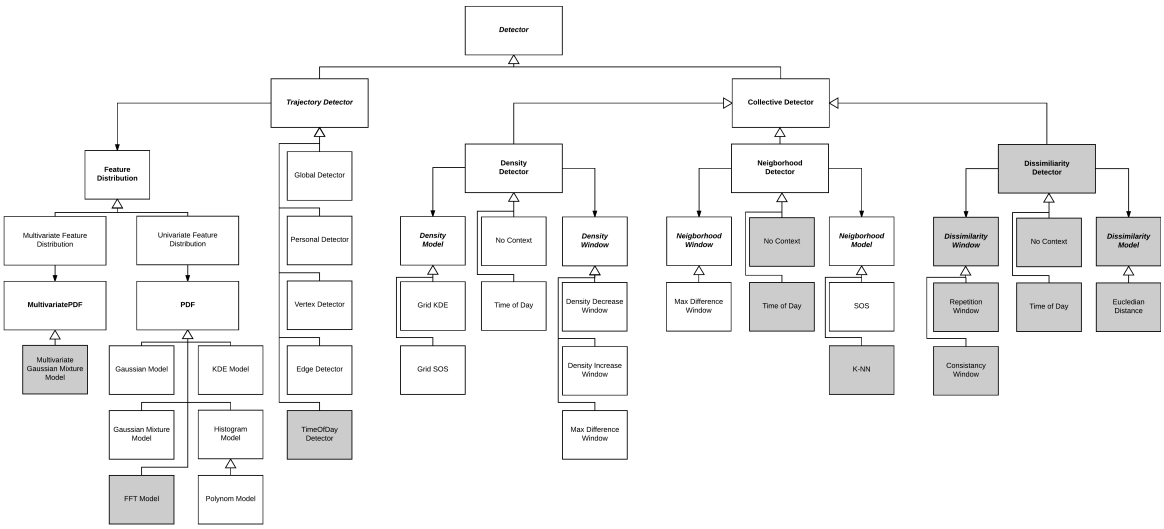


Figure A.3: A diagram of the implemented and suggested anomaly detectors.

Appendix B

Simulator modifications

Two reasons for changes in the simulator: mandatory changes for testing purposes and bug fixes. All bugs mentioned are either reported to and fixed by TNO, or temporary local fixes (hacks) until a better solution is found.

B.1 Export positions

The positions of the agents in the simulator had to be made available for external use such as for the designed detectors. Two methods are added to the simulator to provide this export function. One method writes the positions of all agents to a file, the other streams the positions over a network connection. The first method is mostly used during the research where the last mentioned works well for demo purposes.

To prevent slowing the simulator down, both export methods use a producer-consumer like implementation. If saving the positions cannot be done in real time this will buffer the positions before writing them to file or socket.

B.2 Variation in speed

The simulator had build-in methods to have agents travel with speeds based on gender and age. Unfortunately the actual speed of the agents was hard-coded at $\frac{4}{3.6}$.

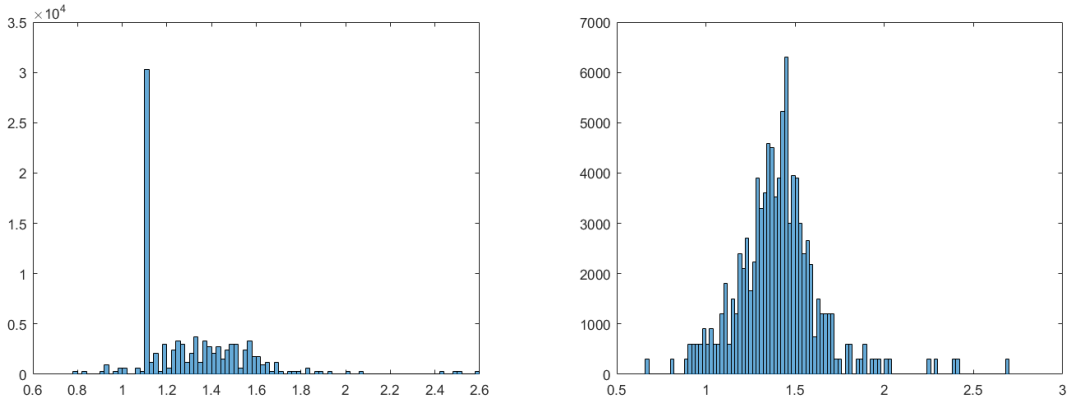


Figure B.1: Histograms of the speed of the agents in the simulator. **Left:** All agents have the same speed. **Right:** The speed is based on gender and age.

Figure B.1 shows how using the build-in method created a more realistic distribution of walking speeds. Note how these histograms differ from the one generated in the detectors. This is due to the sampling of positions. If the position of an agent is sampled while it follows a curve, the distance travelled according to the sampled data is shorter than the real travelled distance. The time taken to travel stays the same resulting in a lower speed in sampled data.

B.3 Simulated events

Simulated events such as street theft and shoplifting were modelled to have the offenders (and victims) walk towards the crime scene. Followed by the crime (which is purely a change or role for the agents). The behaviour following this crime was simulated only when the ”come to aid” or ”arrest” events were used. In those cases either the offenders or victims went to a location where they would meet police or medical personnel. Although these simulations are working fine, one situation was missing: The offenders were walking away like nothing happened. For anomaly detectors, such behaviour is seen as normal and unlikely.

For both street theft and shoplifting, additional running behaviour was added to the simulation. The offenders were modelled to run back to the location they came from, before continuing there routine behaviour.

Appendix C

Statistics

C.1 Theft incidents

Table C.1: Statistics on Dutch Theft incidents.

					2012	2013	2014	2015
Vermogensdelicten totaal	Aantal delicten			per 100 inwoners	20,2	21,2	20	18,9
	Melding en aangifte	Melding		in % ondervonden delicten	46,2	45,1	45	43,7
		Aangifte	Aangifte totaal		36,9	36,3	36,3	34,7
			Aangifte via procesverbaal		25,1	24,3	24	22,1
			Aangifte via internet		11,9	12	12,4	12,7
(Poging tot) inbraak	(Poging tot) inbraak			per 100 inwoners	3,9	4,1	3,9	3,6
	Poging tot inbraak				2,4	2,5	2,4	2,2
	Inbraak				1,5	1,6	1,5	1,4
Fietsdiefstal	Aantal delicten				5	5,5	5,6	5,7
	Plaats voorval	Geografische lokatie	In eigen buurt	in % ondervonden delicten	55,7	55,5	55,7	57,9
			Ergens anders in woonplaats		33,9	32,8	33,9	31,1
			Ergens anders in Nederland		8,1	8,7	8,5	8,7
			Onbekend		2,4	3	1,9	2,4
Autodiefstal				per 100 inwoners	0,2	0,2	0,2	0,2
Diefstal uit of vanaf auto	Aantal delicten				3,6	3,4	3	2,9
	Plaats voorval	Geografische lokatie	In eigen buurt	in % ondervonden delicten	80,4	80	80,1	81,6
			Ergens anders in woonplaats		7,7	8,1	6,8	7,6
			Ergens anders in Nederland		7,5	7,5	7,8	5,8
			Onbekend		4,4	4,4	5,3	5,1
Diefstal andere voertuigen				per 100 inwoners	0,8	1	0,9	0,9
(Poging tot) zakkenrollerij, beroving	(Poging tot) zakkenrollerij, beroving				2,1	2,3	2,1	1,8
	Plaats voorval	Geografische lokatie	In eigen buurt	in % ondervonden delicten	26,8	30,4	30,2	30,6
			Ergens anders in woonplaats		37,7	36	35,8	35,8
			Ergens anders in Nederland		29,6	29,7	30,2	31,1
			Onbekend		3,4	3,3	3,5	4,1
		Pleegplek	Thuis		2,9	4,4	5	4,1
			In een horecagelegenheid		15,1	13,4	12,4	14,4
			In openbaar vervoer		6,8	8,1	7,5	9,1
			Op straat		26,2	26,3	29,2	27,5
			Op werk		4,3	4,7	4,5	4
			Op school		4,9	3,4	2,9	3,8
			In een winkel		20	21,1	20,8	18,5
			Ergens anders		16,5	15,3	14,2	14,4
	Poging tot zakkenrollerij			per 100 inwoners	0,4	0,6	0,5	0,5
	Zakkenrollerij				1,5	1,5	1,4	1,2
	Poging tot beroving				0,1	0,1	0,1	0,1
	Beroving				0,1	0,1	0,1	0,1
Overige diefstal	Aantal delicten				4,5	4,7	4,3	3,8
	Plaats voorval	Geografische lokatie	In eigen buurt	in % ondervonden delicten	55,6	56,6	58,2	55,4
			Ergens anders in woonplaats		23,8	22,6	22,2	25,5
			Ergens anders in Nederland		17,1	17,9	16,9	16,1
			Onbekend		5,6	5,9	5,5	5,4
		Pleegplek	Thuis		37,1	38,1	37,5	37,7
			In een horecagelegenheid		7,4	6,9	5,8	7,1
			Op straat		12,5	13,1	15,1	12,8
			Op werk		5,9	5,1	4,9	5,7
			Op school		7,2	6,8	6,2	6,1
			In een winkel		2,4	2,2	2,3	1,6
			Ergens anders		21,9	22	22,7	23,7

Table C.2: Age of offenders (per 10.000 inhabitants) in The Netherlands in 2015

		Totaal verdachten	12 tot 18 jaar	25 tot 45 jaar	25 tot 45 jaar	45 tot 65 jaar	65 jaar of ouder
Totaal verdachten van misdrijven		113	178	253	156	75	21
Verdachten van vermogensmisdrijven	Totaal vermogensmisdrijven	40	97	85	49	25	8
	Diefstal/verduistering en inbraak	33	83	68	39	21	7
	Diefstal van fiets	3	9	6	3	1	0
	Diefstal van bromfiets/snorfiets	1	7	3	0	0	.
	Diefstal van personenauto	1	1	2	1	0	.
	Diefstal uit/vanaf personenauto	1	2	4	1	0	0
	Straatroof	1	5	4	1	0	.
	Zakkenrollerij	0	1	1	0	0	0
	Winkeldiefstal	15	39	20	17	13	6
	Diefstal/inbraak uit woning	3	6	11	4	1	0
	Diefstal/inbraak uit schuur/garage	1	2	1	1	0	0
Verdachten vernieling en openbare orde	Totaal vernieling en openbare orde	16	43	47	18	6	1
	Vernieling en beschadiging	7	18	17	9	3	1
	Vernieling aan auto	2	3	5	3	1	0
	Openbare orde misdrijven	7	22	24	6	2	0
	Openlijke geweldpleging	5	20	18	3	1	0
	Brandstichting	1	3	1	1	0	0
Verdachten van Geweldsmisdrijven	Totaal geweldsmisdrijven	33	39	67	51	24	5
	Mishandeling	21	27	46	32	14	3
	Bedreiging en stalking	10	10	17	16	8	1
	Seksueel misdrijf	2	3	3	3	2	1
Verdachten van verkeersmisdrijven	Totaal verkeersmisdrijven	21	6	48	33	16	6
	Verlaten plaats ongeval	4	1	9	6	3	3
	Rijden onder invloed	16	3	37	26	12	3
Verdachten van drugsmisdrijven		10	8	27	17	5	1
Verdachten van vuurwapenmisdrijven		3	7	10	4	1	0