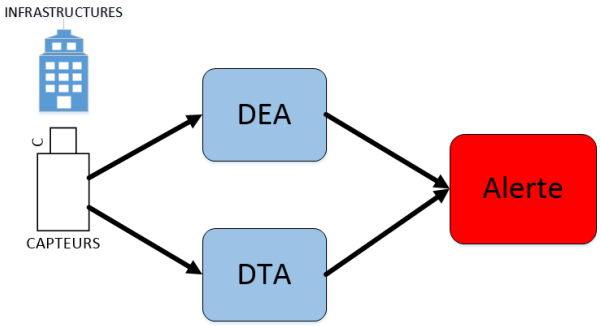
Detection of atypical events for security in critical infrastructure

van Khoa Le

Thèse
de doctorat
de l'UTT

Van Khoa LE

Detection of Atypical Events for Security in Critical Infrastructure



Spécialité :
Optimisation et Sûreté des Systèmes

2018TROY0033

Année 2018

utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SÛRETÉ DES SYSTÈMES

présentée et soutenue par

Van Khoa LE

le 17 octobre 2018

Detection of Atypical Events for Security in Critical Infrastructure

JURY

M. I. NIKIFOROV	PROFESSEUR DES UNIVERSITES	Président
M. P. BEAUSEROY	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. D. TOMASSI	PROFESOR ADJUNTO	Examinateur
M. V. VRABIE	MAITRE DE CONFERENCES - HDR	Rapporteur
Mme S. LELANDAIS-BONADE	PROFESSEURE DES UNIVERSITES	Rapporteur
Mme E. GRALL-MAËS	MAITRE DE CONFERENCES - HDR	Directeur de thèse

Acknowledgement

I sincerely like to thank my honorable Ph.D. advisor, Professor Pierre BEAUSEROY for giving me the opportunity to work with him. I must have to express that he is an amazing Ph.D. supervisor and I am extremely fortunate to be one of his Ph.D. students. My words are powerless to express my gratitude to him. His patient guidance, encouragement, carefulness, advice, and prompt replies to questions and queries have played the important key role in this thesis. He has always provided very concrete ideas and comments throughout my Ph.D. research. I am extremely grateful to him for his enormous support during my Ph.D. study.

I would also like to express my sincere appreciation and my sincere gratitude to my Ph.D. co-advisors Edith GRALL-MAES for guiding me during my Ph.D. research. I am very grateful for her valuable contributions to the publications. Her reviews and comments on the articles were very constructive. I have to say that without her help this thesis would not be a reality.

I would like to thank all my friends and colleagues at the Troyes University of Technology. They were always ready to help me whenever I was in need. They were very talented as well as very friendly, and I had a great time in the group. Thanks also to administrative staff Denise PASCALE and the secretaries of the LM2S laboratory for their valuable supports regarding administrative issues. I also do not want to miss the opportunity to thank my colleagues Ngoc-Tan NGUYEN and Nhan-Quy NGUYEN for whom I spent a big part of my department life as well as social life during my Ph.D.

I express my gratitude to my amazing family, especially my parents, who encouraged me throughout my studies. Everything I have today is because of you. You always be there for me, provide help. Despite the distance, I feel like you are always by my side. Finally, I would like to give special thanks to my girlfriend, Thi-Kim-Ngan LE. Her love, patience, encouragement, supports, and understanding helped me to work better.

Contents

List of Figures

List of Tables

Abreviations

AED Abnormal Event Detection using key metrics.

AP Activity Path.

ATD Abnormal Trajectory Detection.

DBSCAN Density-based spatial clustering of applications with noise.

DTA Dynamic Time Alignment.

DTW Dynamic Time Warping.

ED Euclidean Distance.

FOV Field Of View.

FPR False Positive Rate.

GA Global Alignment.

GDTW Global Dynamic Time Warping.

k-Means k-Means clustering.

KA Kernel Alignment.

LOO Leave One Out.

one-class SVM One-class Support Vector Machine.

POI Point of Interest.

RBF Radial Basis Function.

ROC Receiver Operation Characteristic.

SVM Support Vector Machine.

TPR True Positive Rate.

Abbréviations en Français

CdV Champ de vision.

DEA La détection des événements atypiques en utilisant la métrique clé.

DTA La détection des trajectoires atypiques.

DTW Dynamic Time Warping.

Dynamic Time Alignment Dynamic Time Alignment.

GA Global Alignment.

GDTW Global Dynamic Time Warping.

seul-classe SVM Seul-classe Machine à vecteurs de support.

SVM Machine à vecteurs de support.

Chapter 1

Introduction

This thesis aims to develop approaches for anomaly detection in systems of critical infrastructures. It was carried out in the context of the VIRTUALIS project as part of the FUI15 program. VIRTUALIS gathered ten partners among which the System Modeling and Dependability team (M2S) of ICD laboratory. The objective of this project was to develop an automatic security system for critical infrastructure. Our primary contribution to the project was to propose detection methods for security applications, using data from cameras and sensors. The VIRTUALIS project lasted for three years and started in July 2014, about a year before the beginning of the thesis. The work was mainly done on the base of the project. However, due to a lack of data from the VIRTUALIS project, public data of a project named ATC were also used.

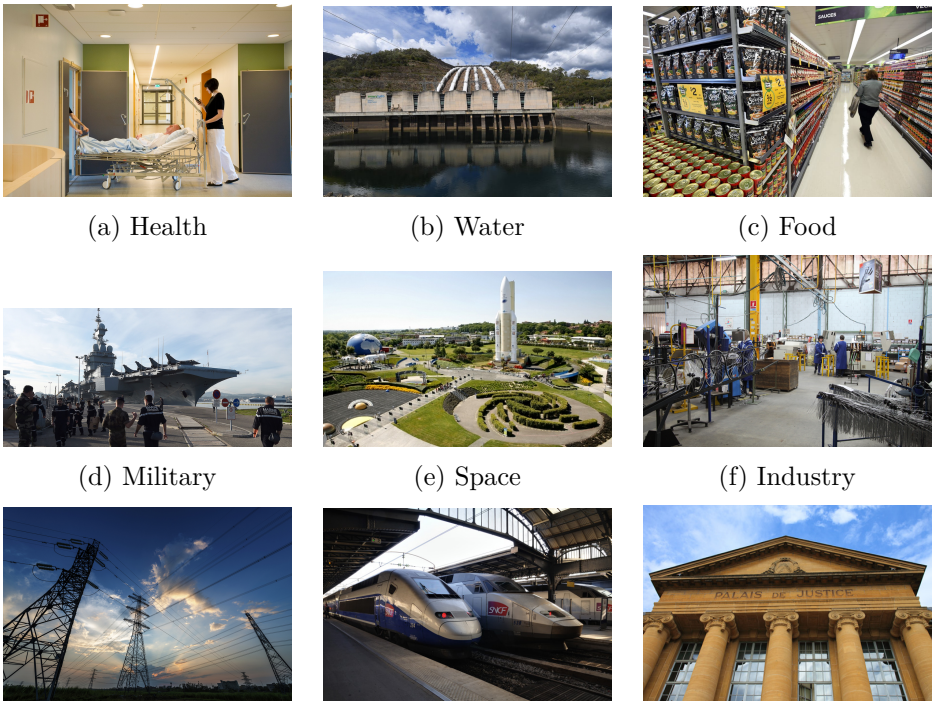
This introduction begins with a brief description of critical infrastructure followed by a description of the current security issues and the motivation of VIRTUALIS. A brief introduction of the ATC project is also mentioned. It ends with a short presentation of our main contribution to the VIRTUALIS project and the outline of the manuscript.

1.1 Critical infrastructure

Critical infrastructures are defined as a network of independent, mostly private systems and processes that function collaboratively and systematically to produce and dis-

tribute a continuous flow of essential goods and services [48]. The definition of essential infrastructure is similar throughout the world.

Critical infrastructures are complex systems [30], and each component of infrastructure constitutes a small part of the intricate web that forms the overall infrastructure. The state of one infrastructure is correlated to the state of the other which create an interdependency network between infrastructures in an extensive system. Critical infrastructure covers public buildings such as airports, train and metro stations, administrations, tertiary buildings (banks, offices) and industrial sites (refineries, nuclear power plants, water purification plants). Nation's critical infrastructures provide the essential services that underpin the society and serve as the backbone of the nation's economy, security, and health. These infrastructures provide people the power which is used in homes, the water to drink, the transportation, the stores to shop, and the communication systems. Some examples of critical infrastructures are presented in Figure 1-1.



(a) Health (b) Water (c) Food

(d) Military (e) Space (f) Industry

(g) Energy (h) Station (i) Court

Figure 1-1 – Critical infrastructure

1.2 Problematic

The risk analysis of the critical infrastructures has drawn much attention from governments due to its vital importance. Both physical and cyber security are equally crucial for risk management of any contemporary critical infrastructure. Physical security aims to protect property and people by ensuring that only authorized persons and materials are allowed to enter various parts of the premises, using video-monitored, physical barrier equipment, access control, etc. All these devices send their data to a monitoring center, where security operators are managing risk at a macroscopic level using a set of procedures. Physical security, although extremely important, is a relatively mature field in which the threats and preventive measures are well understood [96]. Cybersecurity aims to protect the infrastructure information system from intrusions and leaks, using the notion of role management: access to data is authorized according to entitlements related to people by roles. Cybersecurity, however, is relatively new and represents a particular challenge to interdependent infrastructures.

Since 2015, there were 26 attacks in France, which cause more than 200 deaths and 800 injuries. The attacks that caused the most damage are the shooting in the Charlie Hebdo magazine's office in Paris on 7-9 January 2015, the massive shooting on November 13, 2015 at Bataclan during the Eagles of Death Metal concert, and the vehicle ramming on 14 July 2016 in Nice. Those numbers show that France is a target for terrorists and that security threats are real. Beside terrorist activities, data of citizen are also a target of spying activities, a specific example is the Edward Snowden scandal in 2013, where US National Security Agency (NSA) was accused of collecting telephone records by touching the server of big companies like Facebook or Google [114].

Due to the current critical situation of politics and terrorism, the French President, François Hollande, declared the State of Emergency until July 2017 to help keep residents and visitors to France safe. This decision involves more security checks and greater police power and presence. Despite the enhancement of police's force and security protocol, the ability to detect and prevent attacks of police and special forces is still limited. The difficulty comes from different reasons such as the number of threats (hostile troops, terrorist) or the various number of vulnerabilities of critical infrastructure. More than one countermeasure

may exist for a particular infrastructure, or one countermeasure may reduce the risk for many infrastructures. The bottleneck of new security policies are the overwhelming task of human (i.e., operator's longtime working), result in the unreliability of the security system. So an automatic surveillance system is a particular countermeasure that can be applied to many critical infrastructures to reduce security risk. Therefore, the development of such systems is important in the current situation.

The French government has invested in many projects to create a global road-map for the automatic security systems ([15], [40]). The target of these investments is to create different technological bricks developed for the security of infrastructures. The project VIRTUALIS is one of the key bricks.

1.3 VIRTUALIS project

1.3.1 Overview

The VIRTUALIS project is part of a global approach aiming to propose a system of primary management of the physical and logical security of critical infrastructures. The key issues of VIRTUALIS are to make it possible to optimize and to facilitate the risk management for the security system's installation and operation [32].

The objective of VIRTUALIS project is to create an automatic security system for critical infrastructures such as commercial buildings, metro station. This system can monitor the activities of every people on the site, based on information gathered by a large number of sensors, among which are cameras [31], then analyzes the data and gives an alert to prevent the attack before it happens. Such system can assist operators to detect abnormal events more efficiently, therefore increase the effectiveness of a security system.

VIRTUALIS brought together three major industrial groups, an EPIC, two academic laboratories, an association and three SMEs located mainly in Ile de France but also active in Europe. This consortium was made up of known partners in the world of critical infrastructure security. It grouped the various skills needed to carry out the project based on their complementary.

The project was divided into multiple sub-projects, grouped in packages concerned with administration, law, technical tasks. A particular consortium handled each sub-project. The research work presented in the manuscript is related to sub-project 3 - Analysis of global risk (SP3). It is an intermediate step, which contains multiple detection models developed by the SP3 consortium members. Each model uses preprocessed data from multiple types of sensors, prepared by previous sub-projects as the input, and outputs the detected anomalies. An anomaly could be an abnormal event or unusual behavior.

1.3.2 Demonstration scenario

By default, two types of infrastructures, a metro exchange station, and a working office building were chosen as the scenarios for the VIRTUALIS project. Thales site hosts the working office scenario and the Chatelet metro station in Paris hosts the metro station scenario. Due to the plan, systems of sensors were supposed to be set up at Thales site and Chatelet metro station which correspond to two scenarios to collect data in these environments. Then each member trained their detection models on the same dataset and these models would be integrated into an automatic system of security. The result of the project would be displayed at a demonstration technique. Unfortunately, the data acquisition process of the metro station ran into some administrative issues, which led to an abandon of the metro station scenario. As a result, in the final demonstration of the VIRTUALIS project, the proposed system performed in the scenario of the Thales site.

So as mentioned before, the VIRTUALIS project was finished one year before the thesis. So after the VIRTUALIS project, we still have time to experiment our methods in a second case. We looked for a rich, robust dataset which can replace the metro station case. We finally retained the public dataset from the ATC project and use it as a second experimental case.

The ATC project [19] was conducted in Japan, a sensor system was installed in the ATC shopping center in Osaka Bay. The detail information of the ATC project will be presented in section 2.3.

1.4 Thesis contribution

The main contribution of this work consists of two methods for anomaly detection using trajectory data. The two proposed methods are different and somehow complementary. They can be applied together in a security system. These methods are called Abnormal Event Detection using key metrics (AED) and Abnormal Trajectory Detection (ATD). The methods were applied to the VIRTUALIS case (office building) and the ATC case (shopping center). Since each case is different, the methods were adapted to the specific characteristics of each environment.

- AED is a parametric method, based on extracted features. This method analyzes multiple sub-trajectories which are observed in a critical area of the building. It considers the sub-trajectories as events which happen in the area and extracts key metrics from these events. A training step enables to learn the key metrics distribution and to choose the detection thresholds according to operational demand. The thresholds are applied in the detection stage to classify abnormal situations in the area.

- ATD is a method that detects the abnormal sequences of activities. It processes the data to produce sequences of activities, where each activity is inferred from the gathered data. The detection function is trained based on available data. Depending on the available dataset, the classifier training could be supervised using normal data and attack data, or unsupervised using normal data only. The classifier is then applied to predict the label of new sequences.

These methods has been published as two articles ([69],[70]).

1.5 Organization of the document

The manuscript is organized as follow. In chapter 1, we pointed out the definition and the importance of critical infrastructure for a nation. The risk and limitation of current security system give the motivation for the VIRTUALIS project and this work.

Chapter 2 describes the project behind this work and the data used in this manuscript. The description of the project VIRTUALIS includes the consortium, the sensor system, the development of the dataset in relation to a particular abnormal scenario. In addition to these data, the public dataset of the ATC project will be presented.

To provide a theoretical background for our scientific developments, chapter 3 outlines the problem formulation, which is the anomaly detection for trajectory data, and categorize the methods used by other researchers to respond to that problem. We review some existing works, identify the key research gap in order to position our contributions.

Chapter 4 presents the first contribution named *Abnormal Events Detection using key metric*. Chapter 5 presents the second contribution named *Abnormal Trajectory Detection*. In both chapters, the methodology is presented, and the implementation of the two test cases VIRTUALIS and ATC are explained respectively.

The conclusion of this work and the perspective for future development of automatic security systems are presented in chapter 6.

Chapter 2

Project and dataset

2.1 Introduction

This chapter presents the VIRTUALIS project and the application context where we consider to challenge the proposed methods. One of the key points is related to the data, so special attention is put on presenting each dataset. This chapter is organized into two main sections. The first one presents the VIRTUALIS project, its objective, the consortium, the sensor system and the data. In the case of VIRTUALIS, we have two datasets, a simulated one and a real one. Attacking scenarios were prepared by security experts to test the proposed methods. The second section presents the ATC case: the context, the sensor system, and the dataset. Although we did not participate in this project, the sensor system was described very carefully in the papers.

2.2 VIRTUALIS project

With the technology evolution in recent years, the numerical system is more and more important in all applications, organizations and human systems. It implies a specific challenge for security system where physical devices are usually built on one side, and information system security is designed on the other side. There are many reasons to organize the work this way, but it may lead to security leaks, incoherence in processes. So

the main idea of VIRTUALIS project is to propose a global approach aiming to manage the physical and logical security of critical infrastructure. Such an approach makes it possible to optimize and facilitate the risk management for the installation and the use of the security system. This system provides alerts, verifies and validates security procedures. The VIRTUALIS project aims primarily to overcome the following technological limitation of existing system [32]:

1. Heterogeneity and mono-functionality of sensors and their interfaces.

   The physical security management of infrastructure now relies on several heterogeneous networks of sensors [11] (e.g., biometric access control or badges, video surveillance, perimeter protection, smoke detectors, etc.) in which a sensor typically provides a single function (e.g., smoke detection).

2. Existing real-time risk analysis tools not suitable for combining physical and logical security data.

   Many risk analysis tools are now available in various fields [9], for example, the management of the security of information systems (e.g., attack graphs) or in the analysis of industrial risks [122]. However, most of them are not suitable for the project's goal of conducting a joint risk analysis of the physical security and logical security of the infrastructure. In fact, risk analysis is not yet thought globally, while threats combining physical and logical security already exist. For example, the computer worm "Stuxnet" discovered in 2010 by a Belarusian company developing antivirus [44]. Stuxnet was used as part of an Advanced Persistent Threat (APT) and aimed at compromising industrial control systems. In particular, Stuxnet has been able to infect physical infrastructures such as SCADA (Supervisory Control and Data Acquisition).

2.2.1 Virtualis consortium

The VIRTUALIS project consortium grouped companies and organizations which bring expertise in different domains. The project leader was Thales, it had gathered partners with a rich history of collaborations. As part of its critical infrastructure security activities, Thales regularly meets with partners involved in related projects (CEA, RATP,

etc.). The committee in charge of the administrative and technical follow-up of VIRTUALIS consisted of the entire consortium and was chaired by Thales Services SAS. Nevertheless, an Administrative and Technical Steering Committee (CPAT) was set up to monitor specific critical activities and critical decisions. The CPAT was composed of the partners responsible for the primary sub-projects:

- **Thales services SAS**: Thales is a world leader in critical information systems. It has a global workforce of 68,000 people and a presence in 50 countries. Regarding R&D, Thales invests 2.2 billion euros in total (18% of activity), employing 25,000 researchers in cutting-edge fields that produce more than 300 inventions.
- **Sispia SARL**: SISPIA is an innovative company created in 2003, with an activity focused on analysis, algorithms and data processing and knowledge for risk and performance management. The major axes of SISPIA cover the interaction between the components, the divergence of the results and the parameters with particular incidence.
- **EGIDIUM Technologies**: Egidium Technologies provides global safety-security and incident management solutions for sensitive sites and infrastructures. Addressing industrial sites, events, transport infrastructures or urban safety, its solutions manage the data fusion of all security sensors and allow intuitive supervision adapted to the management of incidents and crises.
- **Regie Autonome des Transports Parisiens (RATP)**: The RATP Group is the world's fifth largest public transport operator with a presence in all modes of collective mobility: metro, rail, tramway, and bus. In Ile-de-France, RATP operates, maintains, modernizes and develops one of the densest multimodal networks in the world.

As introduced in chapter 1, the VIRTUALIS project is a project within a road-map to create a unified technological framework for the security of the infrastructure. In this project, different people from several cooperations and companies worked together to achieve this mission. Multiple tasks are divided for the appropriate members who are specialists in their field. The project VIRTUALIS was divided into eight sub-projects as illustrated by the diagram in Figure 2-1.

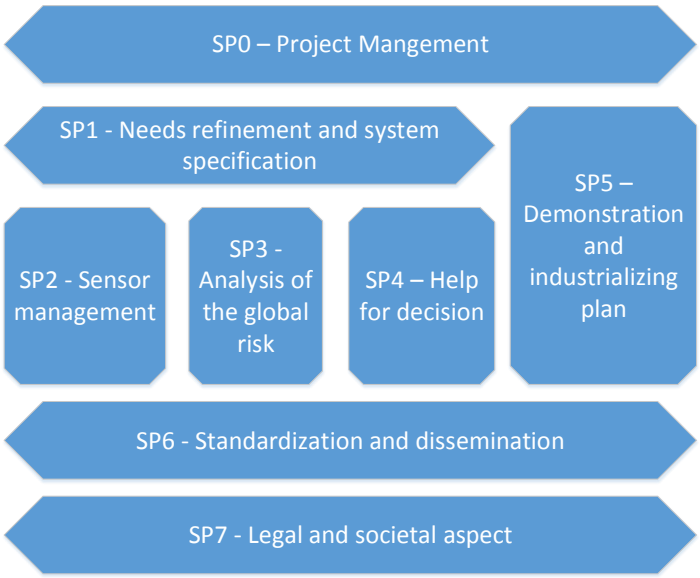


Figure 2-1 – Plan of project VIRTUALIS

The objective of each sub-project is briefly described as following:

1. The sub-project *project management* (SP0), aims to ensure the administrative and technical management of the project.

2. The sub-project *needs refinement and system specification* (SP1) carries out an analysis of the vulnerability risks incurred in the critical infrastructures selected in the project.

3. The sub-project *sensor management* (SP2) aims to enable the system to manage physical sensors according to a multi-tasking approach, by removing the lock of the heterogeneity of sensors and interfaces.

4. The sub-project *analysis of the global risk* (SP3) identifies existing data mining and risk analysis techniques that are best suited to the needs of the convergence of physical security and logical security.

5. The sub-project *help for decision* (SP4) completes the integration of the system to have a functional prototype for the physical and logical security of critical infrastructures.

6. The sub-project *demonstration and industrializing plan* (SP5) demonstrates the capabilities of the VIRTUALIS prototype in a real environment.

7. The sub-project *standardization and dissemination* (SP6) ensures outward communication about the project and the dissemination of results obtained during the project.

8. The sub-project *legal and societal aspect* (SP7) has a mission to integrate the VIRTUALIS project in a global approach of legal and regulatory compliance.

The Systems Modeling and Safety laboratory (M2S) is one of the team members of the Charles Delaunay Institute, the research laboratory of the Troyes University of Technology (UTT). The M2S researchers whose specialist areas cover decision, diagnostic reliability and maintenance by essentially statistical approaches. In the VIRTUALIS project, M2S researchers participated mostly in SP3 sub-project.

2.2.2 Analysis of global risk - SP3

This work was part of the SP3 sub-project. This sub-project, led by SISPIA, aimed to conduct a risk analysis for global security (physical and logical) by dealing with the issue of the inadequacy and the incompatibility of existing tools in the field of security. The aim was to develop the ability to combine data from different security systems to treat them in a comprehensive and unified way. The objectives of this task are:

- Identify risk analysis methods for security systems that are adaptable to the issues of physical and logical security.
- Adapt these methods to the context of physical and logical security.
- Develop a dual physical/logical risk analysis software module.

More specifically, the mission of the consortium of the SP3 task was to propose different methods that analyze data from multiple types of source such as camera, card reader, iris sensor, etc. The output of each method was collected, and a fusion technique, in the final stage, decides if the system should raise the alarm or not. According to the

study provided by the SP1 sub-project, the VIRTUALIS project should be tested on two real cases: office building and metro station. Therefore, the SP3 task was also divided into two experimental cases.

In the first case, the system of sensors and security protocol were set up in an area inside Thales building in Palaiseau by the SP2 consortium, and SP3 consortium proposed their methods for this scenario. The second case of the project VIRTUALIS was defined as a metro station scenario. The dataset of this case was supposed to be provided by RATP, and a system of sensors should have been installed in the Chatelet metro station in Paris. However, during the development period, only data from the Thales building was available, mainly because of data privacy issue and problems of getting anonymous data. So the security system of the VIRTUALIS project only performs on the office scenario at Thales. So the VIRTUALIS dataset considered in this work is restricted to the dataset of the office scenario created in relation to Thales building.

Due to the constraint, LM2S researchers came up with four methods in the project, two of them belong to this thesis. Two methods have been proposed within this thesis. They are called AED and ATD.

2.2.3 Context

In this section, we present the environment of the working office scenario and its characteristic. Office buildings are private and closed to the public. Beyond the terrorist risk, the main risks that companies seek to reduce are acts of sabotage and espionage that jeopardize the assets of the company. Therefore, every person present in the infrastructure or connected to the information system must be authorized. These tertiary buildings are mostly equipped with security systems providing access control, perimeter protection, fire detection and card reader to the information system. These security systems operate independently and are therefore not adapted to the convergence of the two physical and logical worlds.

The Thales site in Palaiseau hosts Thales Research & Technology France as the main occupant. This is an example of the office building context. Several types of technical

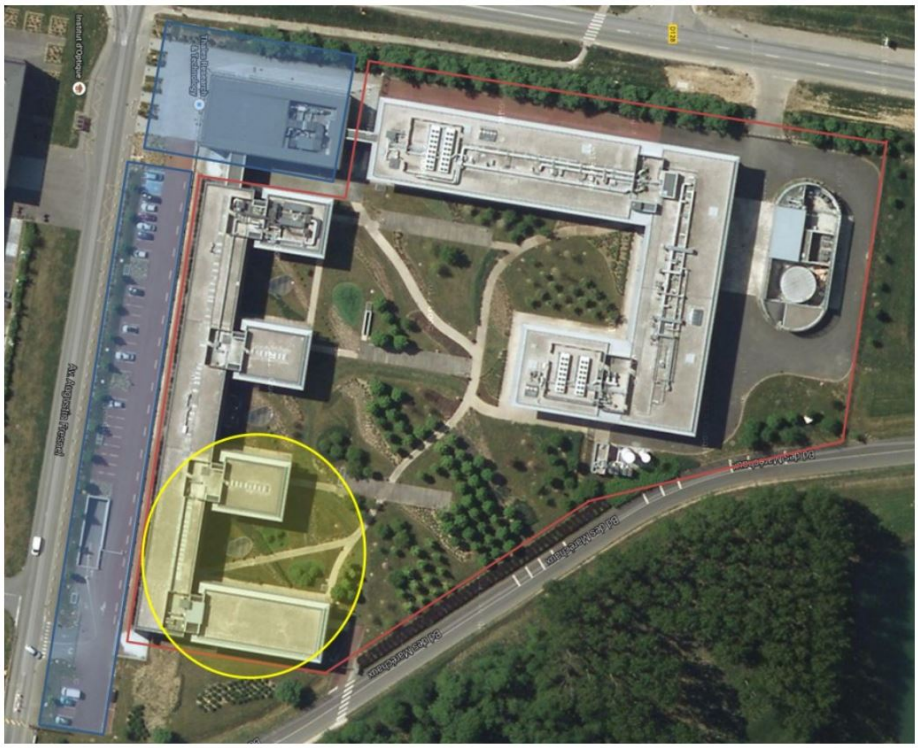


Figure 2-2 – Thales TRT site in Palaiseau, an example of office building.

installations are present on the site: laboratories, offices and other zones in an operational state of support to the objectives of the Research Center. The experiments of the office scenario in VIRTUALIS project was executed at this location.

A view of the studied site is shown in Figure 2-2, where three different areas are highlighted. The areas in blue represent semi-public zones, in particular, the reception of the site and the visitor parking, under the responsibility of Thales, but open to the public; the zones in the red polygon represent the private spaces, accessible by authentication; the yellow circle bypasses part of the site, hosting the ThereSIS laboratory, which is the heart of the vulnerability analysis. This area is the place where the security system of the VIRTUALIS case was installed.

The essential activities of the Thales site include scientific activities, commercial and infrastructure supported activities. Supported services are parallel activities, which were not considered, except for the case of the security service. The scientific activity includes all the technical work done in the infrastructure. The commercial activity refers to the outward communication of the progress of the research activity (demonstrations to

customers, journalists, partners). The security service is responsible for monitoring the site's facilities and premises.

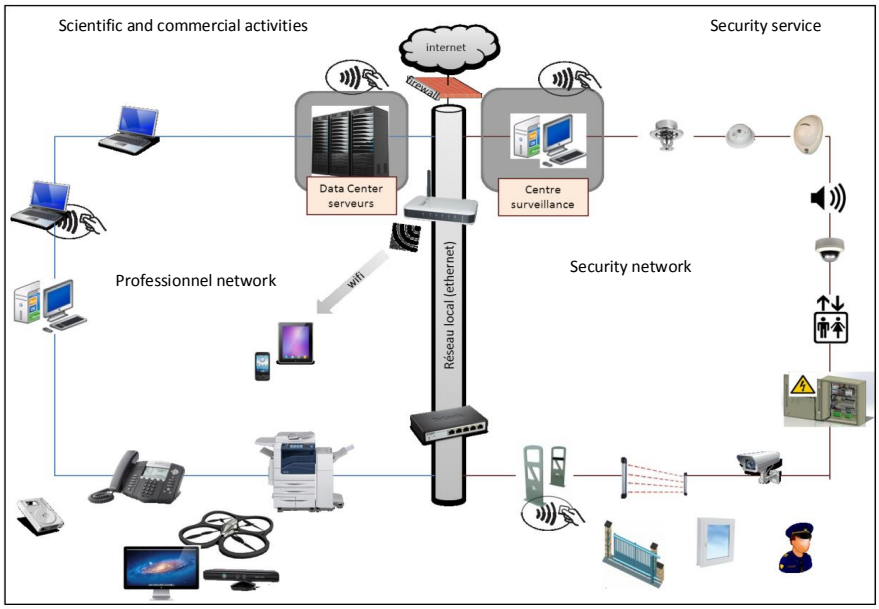


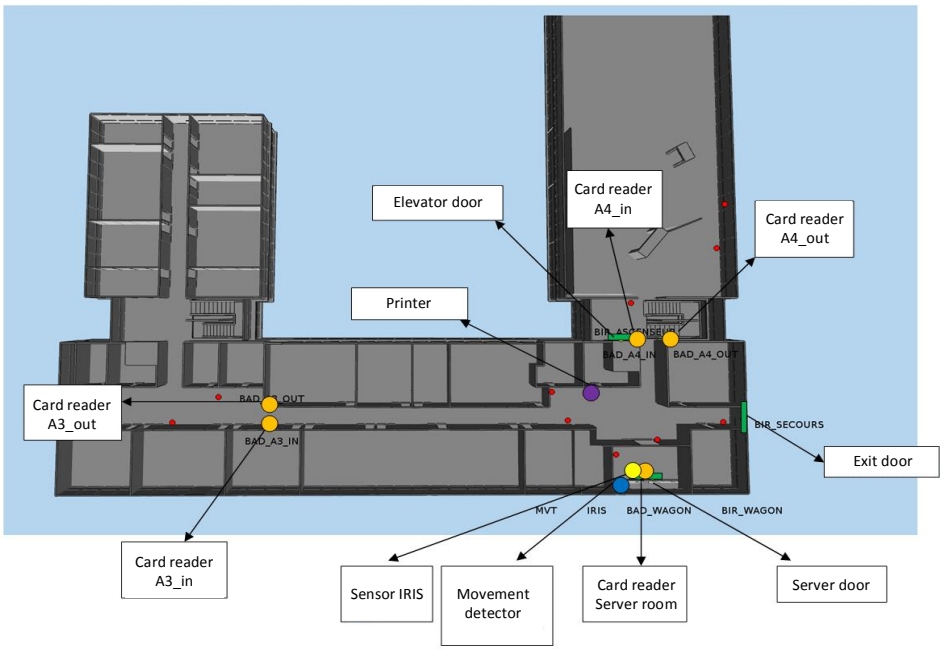
Figure 2-3 – Diagram of Thales TRT site installations

Two categories of activities are supported by two network systems: professional network and security network as in Figure 2-3. The professional network is used to connect multiple types of equipment in the building, for different tasks such as phone call, file transfer, server access. The security network is separated, which is used to control and receive information from security equipment such as gates, elevators, cameras or fire alarm.

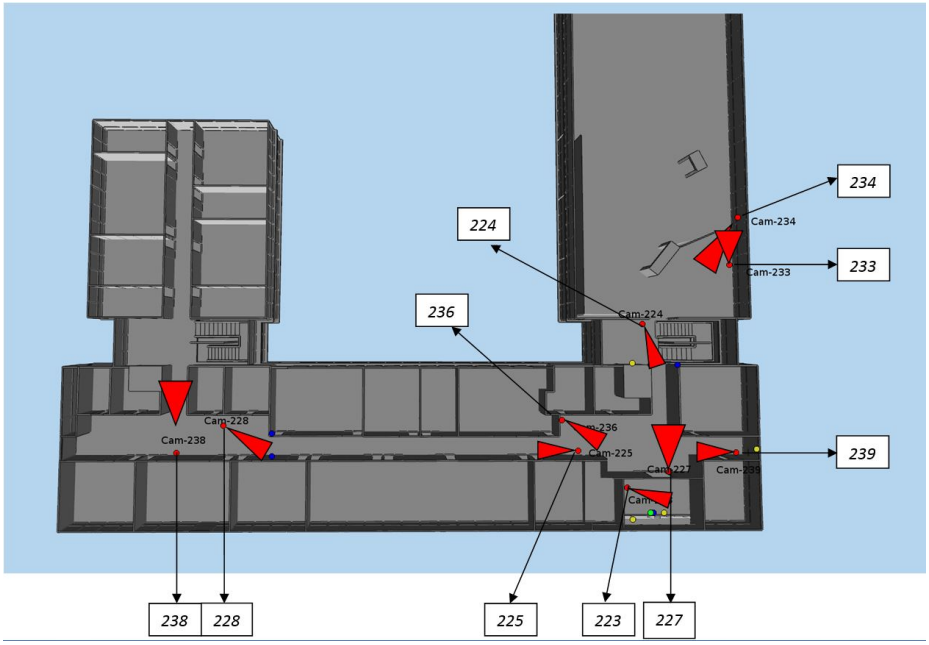
2.2.4 Sensor system

According to the project roadmap, the sensor system was developed by the sub-project SP2, the aim was to provide some processed data so that the sub-project SP3 is able to detect abnormal events.

The sensor system composed of camera and sensors was installed in the ThereSIS area as in Figure 2-4. Multiple types of sensor such as card reader, iris detector, movement detector control every activity in the ThereSIS area. All employees has to scan their card each time they enter and leave the area. There are two gates in ThereSIS area named



(a) System of sensors installed on Thales TRT site



(b) System of camera installed on Thales TRT site

Figure 2-4 – System of sensors on Thales site

A3 and A4 entrance in Figure 2-4. Other card reader control are put at the server door, and at the printer. Only authorized employees are allowed to access these areas. Along with cameras, other sensors were installed to monitor the area. Cameras's distribution is

presented in Figure 2-4b. The camera's Field Of View (FOV) were concentrated on the printer zone (Figure 2-5).

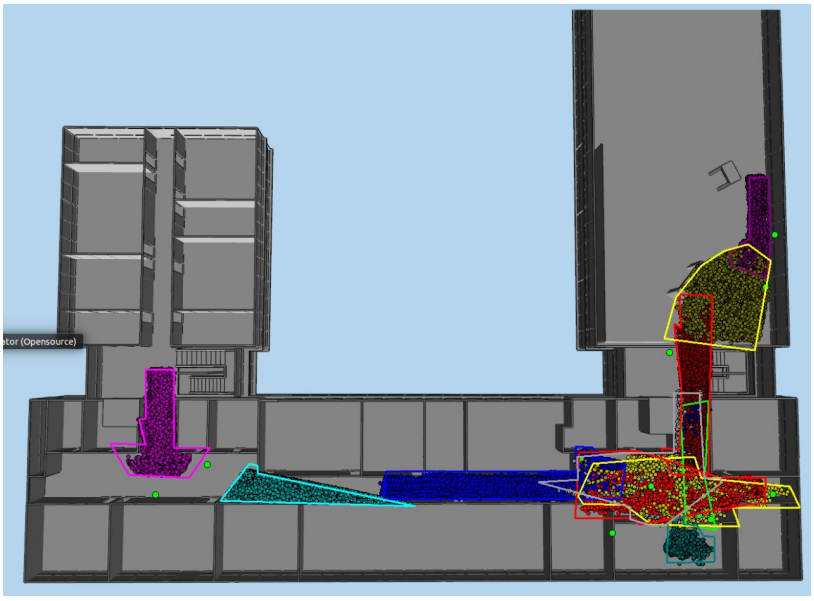


Figure 2-5 – Camera's FOV in ThereSIS laboratory.

2.2.5 Raw data processing

With the sensor system described as above, many issues can be solved, including physical and logical points. We can perform multiple tasks with the sensor system in Figure 2-4. The aim is to produce so called 'virtualization data' from raw data captured from cameras and sensors. The physical issues are presented first:

- **Visual presence detection and tracking** [91]: The visual tracking system makes it possible to locate the pedestrians in critical infrastructure by the analysis of the video streams provided by a camera network. The algorithm is supposed to detect and track the pedestrians seen by the cameras, in real time. The visual tracking function is used to locate moving objects in the infrastructure when it is not provided with a specific localization device. It can be used to report potential intrusions into unauthorized areas.
- **Face detection and recognition** [38]: Face detection is a video analysis function that allows the system to determine the presence of a face in a video stream. Face

recognition uses the frame of the face detected, extracts the features of the face and compares with face patterns stored in a database to identify the person. The fact that in the real scenario, cameras cannot always capture human face because of different reasons such as the pose of the camera, pose of the head, illumination, resolution of the image... The solution of SP2 is to associate a 3D range sensor to each camera. With the 3D silhouette tracker, the person's tracking is more robust. The 3D information (position and speed) allows to predict the position of the person and thus obtain less fragmented tracks. The combination of 2D images by camera and 3D silhouette tracks can perform the tracking and recognition task together.

- **Fusion of multi-camera trajectories** [14]: The multi-camera trajectory merge system makes it possible to associate the different trajectories given by the visual tracking system. This module implements a semi-global optimization algorithm for a graph that searches on a sliding time window for the paths connecting the most likely trajectories. One of the advantages of this method is to optimize the graph very quickly to meet real-time needs. Different criteria can be used to make the association. For example, trajectories can be matched using their appearance but also with geometric criteria such as their geographical proximity.

- **Iris extraction and recognition** [82]: Iris recognition is used to provide accurate identity authentication without a PIN number, passwords, or card numbers. Authentication process takes less than two seconds. This technology is video-based. Although "iris scanning" terminology is often used to refer to iris recognition technology, there is no sweeping involved. This technology is based on pattern recognition. With a proximity sensor-enabled device, a subject positioned 10 to 30 centimeters away is guided by a mirror, audio-assisted, and interactive interface to allow an autofocus camera to take a digital video from the camera iris. Individual images from the live video are captured using a capture card. The innovative iris recognition algorithm analyzes patterns in the iris that are visible between the pupil and the sclera (white of the eye) and converts them into a digital signature. This value is stored in a database and is communicated to the identification control units.

- **Non-visual presence detection** [45]: Perimeter detection systems detect the crossing of a light curtain by an individual. The infrared barriers can do such task. It

consists of emitters generating infrared beams of light and receivers that detect the cleavage of the beams. This sensor allows to analyze the distance of the optical return in order to detect the passage of a person and his direction.

- **Card readers control** [66]: Each staff member of the company is provided with an identification card. The entrances and exits of the staff are recorded by card readers using this card. These recordings are composed of the identifier of the card, the date and time of the passage, the direction of passage and the location of the card reader. The analysis of its cross-recordings with other data (example: staff planning) can be used to identify abnormal behavior and extract information related to the possible misuse of cards.

- **Unauthorized usage of the printer**: Detection of unauthorized usage is based on the principle that a person who sends a print request cannot be in front of the printer at the same time. So the person presents in front of the printer at the time of printing and get the document is, therefore, a person stealing documents or photocopier for industrial espionage purposes. This task is done by combining the presence detection by camera and printing command signals. Therefore a card reader control is attached to the printer, so only the owner of the printing command can withdraw his document.

Unlike the physical aspects, the logical issues in security do not require many equipments. Most of the task is done by software installed on the computer, some of the tasks are listed below:

- **Intrusion detection system** [20]: this issue is solved by implying that any access from an employee's computer to the network means he is inside the infrastructure. A network connection of a computer of a person who has not badged his entry is therefore suspect and potentially falls under the usurpation of access rights.

- **Phishing detection** [61]: Several types of tools can be used to detect a phishing attempt by a site. These include tools that scan all visited URLs and connect to a centralized database that lists all known URLs for hosting phishing sites. Other tools analyze in real time the http content visited and apply heuristics to deduce suspicious sites (invalid certificates, complex URLs, ...). Finally, some tools analyze the emails

received to detect it from the source, emails containing suspicious links to phishing sites.

- **Network Access Controller** The network access control (NAC) is often provided by an appliance mechanism to prevent connection to a corporate network unidentified equipment, or not being in a condition deemed "healthy" (e.g., not having updated antivirus rules for several days).

- **Virus detection**: Virus detection is performed by a Host Intrusion Detection System (HIDS) that works in the same way as network intrusion detection systems, but on a machine. The detection of viruses is done by comparing the files and executables present on a machine to a database of known virus signatures. It can also be done by analyzing the suspicious behavior of applications (repeated system calls, adding new drivers, ...).

The VIRTUALIS project has proposed solutions to perform many tasks in a security system. The physical system installed in the ThereSIS laboratory and the raw data processing step allowed to collect 'virtualization data' considered as an input for the risk analysis methods, which are the key point of SP3.

It has to be noticed that all the sub-projects started at the same time. So while the SP2 encountered many problems in installing the sensor system at ThereSIS to collect real data, or in making agreement on the format of the data, the proposed methods of SP3 were developed. The SP3 consortium needed data to train and validate the models. Therefore, a temporary dataset was required to test the methods, this was mainly the motivation to create the simulated dataset.

2.2.6 Simulated dataset

The simulated dataset was created using software by the Thales services SAS. The simulation system used a building similar to the real one (Figure 2-6). The timestamp of people's activity follows Thales's schedule (starting around 8 a.m, leaving around 18 p.m, lunchtime between 12 p.m and 14 p.m). Some versions of simulated dataset have been

built. The latest version was available in Juin 2016 and covers nine days. The experiments described in this manuscript used the latest version of data.

Figure 2-6 – Simulation of Thales site.

Table 2.1 – Raw trajectory of the simulated dataset.

ID	Time	X	Y
101	9:20:01	4.5	-5
101	9:20:02	4.6	-4.8
101	9:20:03	4.7	-4.6
101	9:20:04	4.8	-4.3

The simulated dataset and the real dataset are both composed of two parts: environmental information describing the position of the offices, elements in the building, and the operational records which contains trajectories from camera and sensor records. These information simulates the data that camera and sensor provide after the processing task. The simulated dataset is very complete. Since the data are generated, the tracking and recognition tasks are assumed to be perfect. In addition, it is assumed that the cameras cover perfectly the monitored area. It means that the precise position of every people in the building at any moment. The simulated environmental information described the camera's field of view and the coordinates of the critical elements. The zone covered by the camera is given in green polygons in Figure 2-6.

The trajectory's data structure is composed of 4 features: Time, X coordinate, Y

coordinate, person ID. It corresponds to each observation at each time stamp for each person. Table 2.1 gives an example of a trajectory of an employee with ID 101.

Also, the simulated dataset also contains additional information such as the log from card readers at many locations like entrance, printer. An example of a sample is shown in Table 2.2. It has to be noticed that the coordinates can be found since the location of the card readers are known. Two supplemental information in this log is the type of action in each location and the functionality of the employee.

Table 2.2 – Simulated data: card readers record.

ID	Time	Action	Location	Function
101	9:20:01	pass	A3 IN	ENGINEER
301	9:25:21	pass	A4 IN	ADMIN
102	10:01:01	print	PRINTER	ENGINEER
103	10:20:01	withdraw	PRINTER	ENGINEER

This dataset allows to test many methods to analyze and detect abnormal events.

2.2.7 Real dataset

Two data acquisition campaigns were organized at Thales Research & Technology's building in Palaiseau, in the ThereSIS area. The first campaign was done from February 22 to 26, 2016 and the second campaign was carried out from November 7 to 10, 2016. The second version of real dataset is used in our experiments.

Environmental information

The real environmental information is different from the simulated one, it is described as below:

- **ID reference**: registered unique ID for each employee in the Thales site and their personal information.
- **Element's position**: coordinate of every element inside the building, from the sensors, doors, elevator to the printer, etc..

- **Surveillance's boundary**: the polygon of camera's FOV in the building. However, the FOV does not cover all areas of the building such as inside the office, and some 'dead' corners on the corridor. On the contrary, the Printer area is covered by many camera's FOV (Figure 2-5).

Operational records

The real operational record holds the records of each activity in the building. It is an enormous dataset with growing amount of data. This dataset includes the raw data of sensors and data after the preprocessing step which is called the virtualization data. These datasets were created by the SP2 members. Since we work on the SP3 sub-project, we are only interested in the virtualization data and our methods are developed based on this dataset. There are four categories of the virtualization data, they are presented as:

- **Access control**: Each observation in this category is created after a login by a card reader or Iris sensor. This category contains the output of the Iris recognition, double authentication iris/card, double identification face/card.

- **Localization**: the localization information is the largest category. It contains the Iris verification results and the trajectories captured from all cameras. This category contains the output of the multi-camera fusion module, visual tracking, face and iris recognition task.

- **Correlation**: the correlation category provides the information to connect discrete events of localization. It connects two instances of localization type that are possibly linked to the same person. It is the most important information to construct a complete trajectory. It contains the output of several modules of the sub-project SP2, includes the multi-camera fusion module, the face and iris recognition, the double authentication iris/card, and face/card.

- **Intrusion**: The intrusion category gathers the information from the infrared door sensor (non-visual presence detection), which captures every event linked to an entrance and an exit for the doors of the server room and of the building.

Characteristics of the virtualized information

The virtualized information is the result of the data acquisition process. The information of the virtualization data is decided by the raw data collection and the preprocessing step. However, the face recognition can only be done when the person stands steadily in front of the camera. On the contrary, when a person passes through the camera's field of view, only the trajectory is recorded. Therefore, there are two types of localization observations produced by the camera. The first type is when the person's face is captured by the camera, this observation return the person's true ID (ID reference), but there is no trajectory (there is only one captured location, the camera's location, because the person has to hold position in front of the camera). The second type is when the person passes through, the trajectory is captured, but the person's identity is not known. The observation in this case stores the trajectory, and a new person ID is given for this person. The SP2 members have faced a difficult issue to correlate these two types of trajectories. So the real dataset contains two types of trajectories: sequences of camera's location of a known user, or small discrete anonymous trajectories. Thus in the real dataset, the re-identification was not done perfectly. It leads to the fact that several trajectories which should have been associated were not.

The re-identification issue is one of the most complicated problems in a surveillance system. The re-identification requires that a person previously seen by a camera is seen later by another camera and is identified as the same. The difficulty of this task comes from the appearance changes across camera views due to the high difference in illumination, viewpoint or camera characteristics. Most of the current approaches rely on the appearance-based similarity between images to establish correspondences ([12], [27]) but the result is still limited.

2.2.8 Attacking scenarios

As the system of sensors captures activities in the building, data were collected to train the proposed methods. These data contain the normal activities of every people in the Thales building. However, the proposed methods cannot be tested with only normal

activities in the dataset. Therefore, attacking scenarios were prepared by security experts to test and evaluate the performance of the proposed methods in the SP3 sub-project for the abnormal detection task.

These scenarios simulate attacks from concurrents or enemies who want to steal confidential information from Thales's database. Attackers could take advantage of the equipment connected to the professional network (Figure 2-3) like printers, telephone or PC to access the server. So the attacking scenarios were created, by standing on the attacker's perspective, trying to access the critical elements such as a printer and a server room. These attacking scenarios were built for both simulated and real datasets with a small difference between them.

For the simulated dataset, the attack was carried out along in two days. In the first day, a competitor was invited by Thales to a meeting. This person approached the printer, plugged in a USB with malware and stick a note "Do not touch". In the afternoon the same day, another hacker disguised as a printer maintenance engineer and accessed to the information system through the printer. In the second day, a hacker used a card copied from an administrator to access the server room, stole the information and left the building.

The real dataset provided the same attack but with more details. Some actors in the consortium were chosen to play the role of the attackers. In the first day, things happen the same way as in simulated dataset, where attacker attached a USB into the printer to hack into the system. In the second day, a hacker stole a card from an administrator, made a copy and returned the real card. Then he waited for the administrator to leave the building, he used the copy card to access the server room, but the authentication failed due to the double card/Iris check. He returned there for the third day, waited for an administrator to leave the server room, hold the door and entered the server room. After staying in the server room for 5 minutes, he left the room and exited the building.

Until this point, this chapter has described the result of the first two sub-projects SP1 and SP2 of the VIRTUALIS project, including the sensor system in ThereSIS laboratory, the simulated datasets, real datasets and attacking scenarios. These results were transferred to the SP3 consortium to develop abnormal detection methods. The next part of this chapter introduces the ATC project, corresponds to the shopping center scenario.

2.3 ATC case

The VIRTUALIS project started in July 2014 and lasted for three years. During this project, the dataset from RATP was not provided as planned, so we lacked resource and support to achieve our thesis's objective and to evaluate the proposed methods. In that situation, we searched for a dataset that provides real data of pedestrian trajectories in public place. After many days of searching and analyzing different available public datasets ([81], [103], [127]), we selected the ATC dataset [19]. This dataset came from the ATC project that took place at the ATC shopping center in Osaka, Japan. It is large because authors have set up a sensor system to collect data for 92 days, two days a week, spreading in one year from October 2012 to November 2013. As part of a project on enabling mobile social robots to work in public spaces [18] (project homepage in Japanese, founded by JST/CREST), a tracking sensor system was set up in the ATC shopping center. The system consists of multiple 3D range sensors, covering an area of about $900m^2$.

The ATC shopping mall is characterized as a wide surveillance area, large diversity of visitors and is an open environment. Unlike the office scenario in the Thales building, where cameras are only installed in the corridor and camera's field of view is limited inside the office, most of the area in the ATC building is covered by the camera's system, and pedestrians' trajectory is fully captured from the moment they enter until they leave. Although the objective of this dataset is not security, the quantity and quality of trajectories in this dataset attracted our attention, and we have decided to use this dataset to evaluate our proposed methods.

2.3.1 Context

The Asia & Pacific Trade Center, located within the Cosmo Square area on Sakishima (Nanko) Island, is both a business and a pleasant venue. A world-class international wholesale trade complex filled with shops and extensive business information.

In the ATC site, the setup plan of three types of sensors is described in Figure 2-7. This building is a crowded place, and connects a train station, commercial buildings, and a dock. So the pedestrians in this building have different purposes and destinations. In

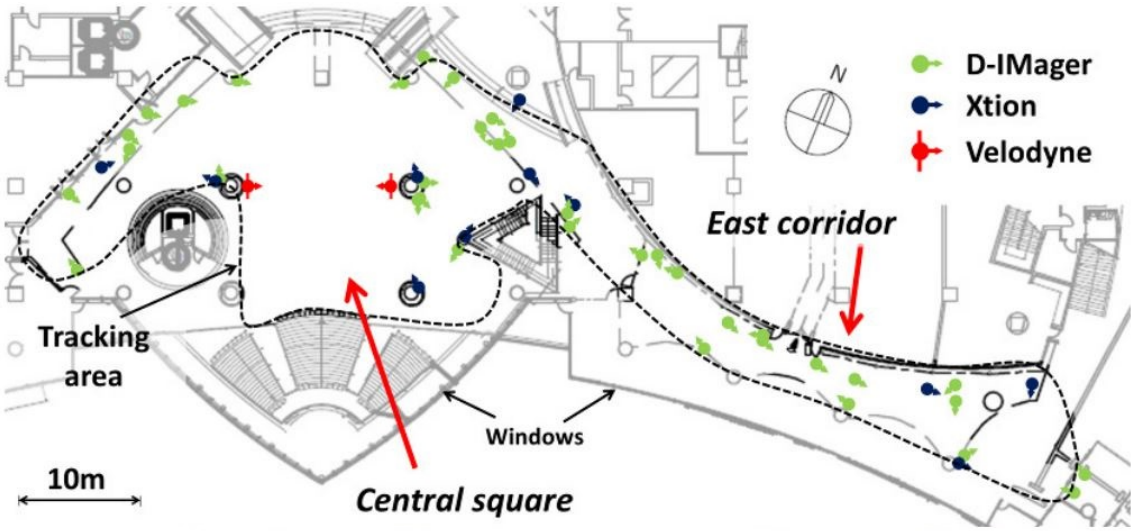


Figure 2-7 – Sensor setup plan in the ATC shopping mall center.

general, the building is divided into a central square and a corridor as in Figure 2-7. The central square is where people stay to wait for the train or ship, and is also a place for exhibition. On the other hand, the corridor is narrow, and people mostly pass through this area.

Unlike the VIRTUALIS project where multiple security tasks are performed in the ThereSIS laboratory, the sensor system in the ATC project focused on pedestrian detection and tracking task. Thus, the two projects are different in their original objective, and in their environment. It is difficult to deploy card readers at the entrances in such context, and there are no critical elements like printers or server.

2.3.2 Sensor system

The sensor system of the ATC project was described by the authors in [19]. The objective was to observe the motion of all the pedestrians in the building, using multiple stationary sensors distributed in the environment. In other words, authors tried to find a practical way to track people in wide public environments automatically for a long period. The proposed requirements of a tracking sensor system are that it needs good tracking accuracy for a varying number of people in the surveillance area, and it is scalable to wide areas and could be connected easily.

The current technology for the tracking task includes the camera, laser range finder, and 3D range sensor. The camera was used in most of the surveillance application [77], as in the VIRTUALIS project. However, there exist the limitation in tracking the position of the pedestrian and the body direction estimation problem [90]. In addition, the implementation of the camera for a wide public place is limited, due to the changing illumination issue. Laser range finder, which uses a laser ray to scan the area has been used for human position tracking in public space ([51], [47]). However, This kind of sensor only perform well in a situation with a small number of people, and it can lead to increasing errors in a wide environment. Many works have applied 3D range sensors for detecting and tracking task ([117], [104]). This kind of sensor can detect the position of the person and the angle of the body and the head.

Due to the environment of wide public space of the ATC shopping center, authors preferred the range sensor over the color camera. The advantage of the range sensor is that it is not affected by the change of illumination, and it is more suitable for the wide and crowded area. However, it contains less information than the color image which leads to a limitation in the recognition task. However, in the ATC case where the research had to respect the privacy issues of the pedestrian. Combining the discussed reasons, authors chose 3D range sensors for the solution of tracking issue in the ATC project. Three types of 3D range sensors were chosen for the experiment:

- **Time-of-Flight cameras**: ToF cameras measure the time a projected light needs to travel to an object to determine the distance. Their usable range is similar to those of structured light cameras, but they, in general, have much noisier measurements, with the level of noise increases with the distance. There is also a substantial interference between sensors which can result in unstable distance measurements. However, the influence of external light for some of the commercially available ToF cameras is much smaller. The price is in the mid-low range, usually several tens of times more expensive than the available structured light cameras. The sensor used here is Panasonic D-IMager EKL31053.

- **Structured light cameras**: This type of sensor measures ranges based on camera views of a projected light pattern. From installation in wide public spaces, these

sensors have a relatively small usable range: the maximum range at which correct and stable measurements can still be obtained is around 5 meters. The measurements are accurate with low noise, especially for close ranges. However, they suffer from a strong influence of external light and from interference between sensors, which can cause missing measurements. The number of missing measurements increases with distance, especially for dark objects like dark hair. The current market price of this type of sensor is very affordable. In this work, authors used Microsoft Kinect1 and Asus Xtion PRO2 models.

- **Multi-layer laser scanners**: These sensors use multiple laser scanning units rotating together to obtain a 3D range measurement. The maximum measurement distance for these sensors is much larger than for above types. They are also very accurate with low levels of noise, and virtually no influence from external light or other sensors. Their main weakness is the low resolution of the measurements in the vertical direction as defined by the number of sensing units, which gives a limit on the range up to which they can be used for tracking. The cost of the sensor is considerably higher than for the above two types. The Velodyne HDL-32E4 were used in the implementation.

These types of 3D range sensors were distributed in the ATC building at the position described in Figure 2-7. They were used to collect data on pedestrians in this location.

2.3.3 Real dataset

The data acquisition was done every week on Wednesday and Sunday, from morning until evening (9:40-20:20). The objective is taking a sample working day and a weekend, to limit the volume of the dataset but keep the information. The dataset was provided as CSV files, each row corresponds to a single observation of a person. An observation is described in Table 2.3. The precise environmental dataset is not available in the papers. Thus, the boundary of the ATC building was estimated from the boundary of the cloud movement points.

The ATC dataset has more features from tracking task than the VIRTUALIS dataset due to the ability of the 3D range sensors. The height of pedestrian is estimated so it

Table 2.3 – Raw trajectory of ATC dataset.

Feature	Description	Example
time (ms)	time of the observation, the time UNIX format is used with a resolution milliseconds.	1353202633.801
person id	identity number given to each trajectory	10370300
X, Y coordinate (mm)	coordinate of the person in a predefined space	-326 2054
Z coordinate (mm)	height of the person	1622
Velocity (mm/s)	speed of the person at that instant	912.316
Angle of motion (rad)	the moving angle of the person compared to the horizontal axis	-1.375
Facing angle (rad)	the angle between face and shoulder	0.548

is possible to predict if this person is a child or a matured person. The speed at each observation is also kept. The angle of motion feature keeps the moving angle of the person compares to a predefined axis. Moreover, the facing angle is the angle between the face and the body, it can tell if a person is looking around or going straight. Although the dataset is only stored for two days a week, its volume is enormous and can be difficult to import and process on a personal computer. The ATC dataset corresponds to approximately 40 gigabytes of compressed files, which takes a lot of space in the memory. So in the experiments, we usually process on a subset of this dataset.

There is a small labeled dataset which was made for an experiment in [18]. It includes trajectories of 29 people who were labeled as atypical by a human coder, as well as 29 people who were labeled as typical. The labeled dataset contains trajectories from two days: 11 and 18 November 2012. In this context, the meaning of anomaly is a person who seems to get lost and need help.

2.4 Conclusion

This chapter presented the sensor system, the environment and the dataset of two studied cases VIRTUALIS and ATC. The thesis is a part of the project VIRTUALIS, in which we proposed the methods to detect abnormal behaviors for security infrastructure. The ATC dataset was chosen as an additional experiment case to apply our proposed methods. This is a public dataset which contains several pedestrian trajectories captured in one year. Although the objective of the ATC project is not for security, the ATC dataset

is still a perfect case to apply our detection methods due to its open space environment and the complicated flow of pedestrians.

The VIRTUALIS project uses various types of sensors and camera to capture all types of activities in a building, from physical to logical. On the other hand, the ATC project performed a robust solution for the tracking task using 3D range sensors. These two projects create an overall view of the security solution for critical infrastructure.

The provided datasets thanks to these sensor systems are spacial-temporal sequence data. The main objective of the VIRTUALIS project, was to develop anomaly detection methods for security in infrastructures. In the next chapter, the state of the art of the anomaly detection problem, when dealing with sequence data is introduced.

Chapter 3

Anomaly detection for trajectory data

3.1 Introduction

The detection of abnormal trajectory is the core issue of this thesis. Different from trajectory patterns mining where the patterns that frequently occur in trajectory data are searched, trajectory outliers detection consists to isolate trajectories or segments of trajectory that are significantly different from others regarding some similarity metrics. One can also be interested in events or numbers of observation that share same properties that do not conform to expected patterns (e.g., traffic congestion caused by a car accident). Discovering abnormal trajectories in a large-scale dataset also has remarkable meaning and practical value and can be associated to the same group of problem.

This chapter aims to introduce the methods and application of the anomaly detection that were mainly applied to sequences. Section 3.2 presents first the general definition of anomaly detection problem, and second, some popular methods for anomaly detection. We categorize the anomaly detection methods into two groups: density-based methods and boundary-based methods. In section 3.3, the definition and approaches using trajectory as input, as well as the methods to process this type of data are described. These approaches motivated us to propose new methods for the VIRTUALIS project.

3.2 Anomaly detection method

Anomaly detection is widely used in various applications such as credit card fraud detection, cyber-security, health care, fault detection in critical systems. Many methods have been proposed to solve this problem in general. Most of them imply a process of three steps: preprocessing, training and detection as shown in Figure 3-1.

The preprocessing stage extracts the data collected from the environment into an appropriate format for the detection model. Training stage trains the detection model depending on the input data and the desired result. Detection stage uses the trained model to give a prediction for new observations.

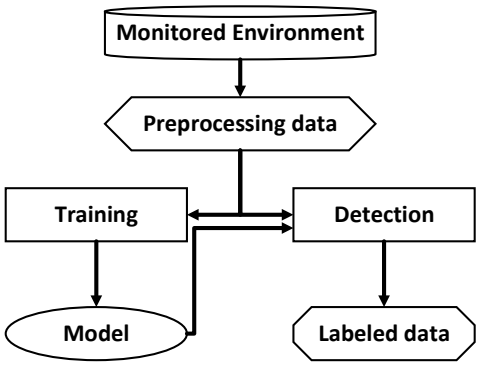


Figure 3-1 – Methodology of detection anomaly

To study more deeply about anomaly detection method, first, we need a precise definition of anomalies. A nice definition has been given by Chandola et al. [25], and adopted by many researchers: *Anomalies are patterns in data that do not conform to a well predefined definition of normal behavior.* However, in many cases we can not well define the normal behavior. So in our point of view, anomalies are also instances that do not conform to the majority samples in the dataset. Anomalies exist in many applications for different reasons, however, they have similar characteristics, and we can categorize the anomaly detection problem into smaller, more specific categories. In fact, by defining different type of anomalies, the input data and the desired output, the anomaly detection problem could be separated in different, more specific problems listed below:

- Based on the output data, one can distinguish:

  - **Scores**: anomaly score of data samples are available. We may want to train the score instead of the decision. A threshold of anomaly score could then be defined to take the final decision: anomaly or not.
  - **Labels**: normal or abnormal labels are given for training samples.

- Based on the input data, the anomaly detection problem is categorized into supervised and unsupervised [49]:

  - **Supervised**: training data have labels for both normal and abnormal classes. It is a particular case of classification problem, where there is only two class normal and abnormal. In a real application, this type of dataset is not obtained easily.
  - **Unsupervised**: training data have no labels, there could be normal and abnormal samples in the dataset, but we assume that the normal samples outnumber by far abnormal ones. This type of dataset is quite popular in anomaly detection problem.

- Based on different type of anomalies as defined in [25] and [86]:

  - **Independent anomalies**: abnormal data sample that appears alone and which is different from the rest of the dataset.
  - **Contextual anomalies**: a sample which is considered as abnormal only in a specific context, also called a conditional anomaly [116]. For example, the temperature minus one Celsius degree in France is normal, but the same temperature in July in Paris is abnormal.
  - **Collective anomalies**: a collection of data instance is abnormal with respect to the entire dataset [52]. Each sample itself is not abnormal, but a group of them becomes abnormal. A typical example is a sequence, where each event of the sequence is normal but the chain of such events could be an abnormal sequence. For instance, Mr.X can be in Troyes at 12 p.m, he can be in Paris at 12:05 p.m. But both these events can not happen in the same day.

The general approach of anomaly detection methods is defining a boundary between the normal instances and abnormal instances in case of supervised learning, or defining the

boundary of a region that contains the unlabeled instances (majority are normal) in case of unsupervised learning. Then this region is used to give labels to new instances: normal label when the sample belongs to the region, abnormal otherwise.

However, several factors make this approach challenging such as the samples near the boundary could be miss classified, the notion of anomaly changes for different application domains, and the lack of abnormal samples for training. Due to the mentioned challenges, the anomaly detection problem is difficult to solve, and it is difficult to evaluate the performance of the detection model.

Many concepts from diverse disciplines such as machine learning, data mining, information theory have been applied to specific problem formulations of anomaly detection [25]. To present the state of the art methods in anomaly detection domain, a detail survey in [25] has been used as a key reference to define the problem and categorize the methods. Latest survey as [3] and [86] have adopted the same definitions of anomaly detection as [25], but they categorize methods differently.

In this thesis, we separate detection anomaly methods into two main categories: density-based methods and boundary-based methods. The density-based methods use the distribution of the data and the boundary-based methods use the instances which lie on the edge of the data distribution.

The rest of this section presents both type of methods mentioned above and introduces the representative methods of each group. However, these methods can not work without a similarity function to compare instances in the data space, and in some situation, the kernel methods are applied to help transfer the data in data space to a high dimensional space so solve non-linear problems. Therefore, the similarity function and the kernel methods are also discussed. Note that the two categories covered in this chapter do not cover all the methods for anomaly detection. However, most of the existing work can be organized using these categories.

3.2.1 Density-based method

Density-based methods use the distribution density estimated from the training dataset to create the detection boundary. In general, high density region has higher chance to correspond to the normal class than the regions that have low density or no training data. Two main approaches in this group are clustering-based methods such as k-Means clustering (k-Means), Density-based spatial clustering of applications with noise (DBSCAN) and the parametric methods like Gaussian model.

- **Clustering-based methods** Clustering-based methods are unsupervised methods, where they try to learn the patterns in the data, and then organize dataset into categorizes with same pattern. Clustering is the action of dividing data into groups of similar objects as in Figure 3-2. Each group consists of elements which are similar according to some specific features [13]. Basically, clustering-based methods are used to categorized unlabeled samples into different clusters. However, some works used these methods for anomaly detection task ([73], [25]).

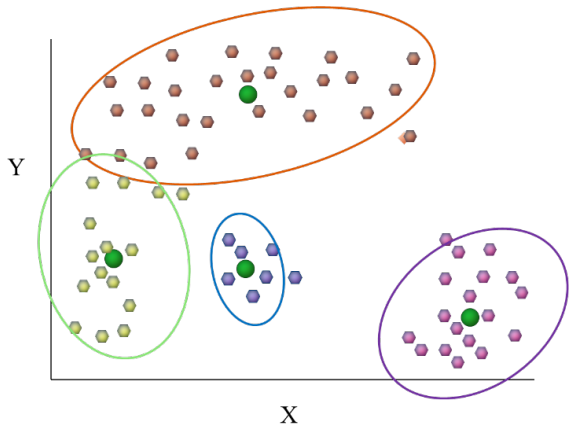


Figure 3-2 – Clustering based methods

Clustering-based methods usually have two main steps. The first step is applying a clustering algorithm on training data. The second step identify the anomalous by a predefined definition about the abnormal samples. For example, abnormal instances could be the smallest cluster, or the instances that do not belong to any cluster. The most popular methods in this group are:

- **k-Means**: Given a number of clusters, this method find samples that belong to each cluster. The only hyperparameter in this method is number of clusters k. The basic idea of this algorithm is that k centers is initiated in the data space. At each iteration, each sample in the training set is assigned to the closest center, then the centers are recalculated. This process is repeated until the centers converge. Many approach using k-Means are introduced for anomaly detection application such as traffic anomaly detection [80], or intrusion detection system [95].
- **DBSCAN**: clusters are identified by looking at the density of instances. The region with high density of instances depicts the existence of clusters whereas region with a low density of instances indicates noise or anomalies. The hyperparameter of this model is the ϵ (maximum radius of the neighborhood) and minPts (minimum number of points must be inside ϵ). The basic idea of the algorithm is that for a given point, if the number of points in ϵ-neighborhood (density) is higher than MinPts, this point is then considered part of a cluster. There are many applications of DBSCAN technique for anomaly detection such as wireless sensor network [41] or in-home activities [56].

Kanagala et al. [65] have compared the advantage and disadvantage of these techniques. In fact, k-Means algorithm could only be applied when the mean of each cluster could be defined, this method produces quality clusters when using huge dataset. DBSCAN can find clusters of arbitrary shape, determine noise and outliers. In DBSCAN, the user has the responsibility to select the hyperparameter values (ϵ and MinPts). A slightly different hyperparameter setting may lead to different clusters. Some difficulties in distinguishing separated clusters appear when they are located too close to each other, even though they have different densities.

The advantage of the clustering-based methods is that the test phase is fast since the number of clusters is small, compared to the k-Nearest Neighbor method where one has to calculate the similarity between the test sample and every samples in the training set. The disadvantage of the clustering-based methods is that they are highly dependent on the clustering algorithm performance and the similarity measure.

- **Parametric methods**

The parametric methods assume that normal data is generated from a parametric distribution with parameter θ and probability density function $f(x, \theta)$, with x is an instance. The base idea is using the training data to estimate the parameter θ. Then use this θ to define a threshold or a criteria. The test instance will be compared with this threshold to decide its abnormality. Two parametric methods are introduced:

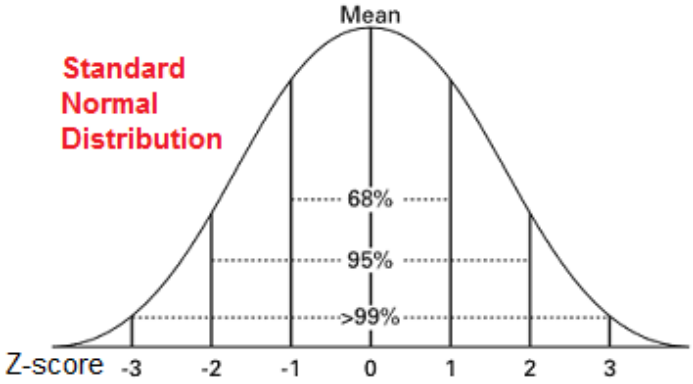


Figure 3-3 – Standard normal distribution

– **Gaussian model based**: this method assumes that each instance is generated from a Gaussian distribution. The hyperparameters of this distribution are mean (μ) and standard deviation (σ). They are estimated by using the Maximum Likelihood Estimation (MLE) method. Given a set of samples x and a statistical model such as a family of Normal distributions $\{N(\mu, \sigma^2)|\mu \in \Gamma; \sigma \in \Sigma\}$. The optimal $\hat{\mu}$, $\hat{\sigma}$ is estimated by maximizing the likelihood function $\mathcal{L}(\mu, \sigma; x)$:

$$\hat{\mu}, \hat{\sigma} \in \{\arg \max_{\mu \in \Gamma, \sigma \in \Sigma} \mathcal{L}(\mu, \sigma; x)\} \tag{3.1}$$

With the estimated $\hat{\mu}, \hat{\sigma}$, the Gaussian distribution function that best fit the training set is established. From this function, a specific criteria is defined, and a threshold is specified for this criteria. For example, the Fisher z test hypothesis [119], where the criteria is the z-score. In the detection phase, the z-score of an instance is compared with the threshold z-score to decide its class. Figure 3-3 presents the standard normal distribution, where $\mu = 0$ and $\sigma = 1$.

– **Mixture of parametric distributions based**: this method uses a mixture of parametric statistical distribution to model the data distribution. Two categories exist in this method. In the first category, methods model the normal instance

and abnormal instances as separate parametric distribution [1]. While in the second category, techniques model only the normal instances as a mixture of parametric distributions ([2], [25]). An example is the work in [43], where an instance x_i is either generated from the majority distribution $\mathcal{M}$, or from the alternate distribution $\mathcal{A}$. The mixture distribution $\mathcal{D}$ is written as:

$$\mathcal{D} = (1 - \lambda)\mathcal{M} + \lambda\mathcal{A} \tag{3.2}$$

They measure how likely each element is an outlier by comparing the difference change in the log likelihood of the distribution if the element is removed.

Many applications using the parametric method were presented for different types of task such as anomaly detection ([102], [62], [93]) or modeling [121]. The advantage of the parametric methods is its simplicity, and the cheap computation in detection phase. However, in the real world, the data may not fit any specific distribution, and this would cause a lot of false detection.

3.2.2 Boundary-based method

Based on training data, the boundary-based methods build a detection function that defines a boundary between normal and abnormal samples. The detection function is computed using the instances near the boundary between the normal and abnormal classes. On the contrary, unsupervised methods construct this function by creating a boundary around the unlabeled (but mostly normal) instances . The principal methods of this group are the Support Vector Machine (SVM) for the supervised classification and One-class Support Vector Machine (one-class SVM) for unsupervised classification.

- **Support Vector Machine**

  The SVM technique was developed by Cortes et al. [33]. This is one of the most popular technique in machine learning for supervised classification. There are many researches that applied SVM methods for different kinds of task like anomaly detection ([46], [55], [67]), fraud detection [120], and intrusion detection system (IDS) [20].

Suppose that we have a set of data samples $X = [x_1, x_2, ..., x_n]$. Each element x_i is a vector with M features, $x_i \in R^{1\times M}$. This set comes along with its labels set $Y = [y_1, y_2, ..., y_n]$, $y_i \in \{-1, 1\}$. SVM is a linear classifier that find the boundary that maximize the margin between two classes of separable data. The optimization problem of SVM is presented in equation 3.3.

$$\begin{aligned} min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ s.t. \quad & y_i(w^T x_i + b) \geq 1, \quad i = 1, ..., n \end{aligned} \tag{3.3}$$

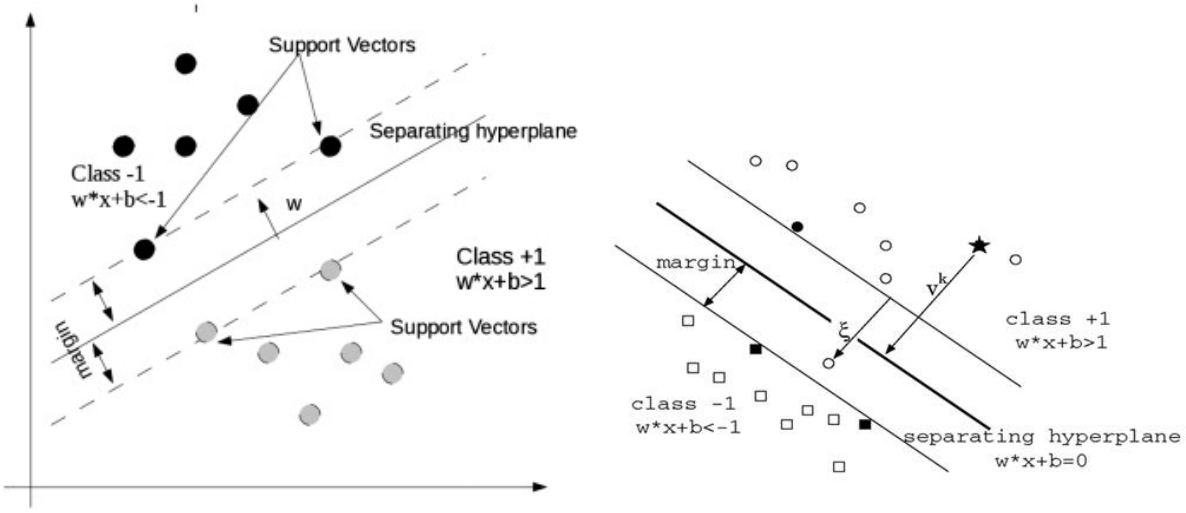
This classifier has a weight vector w and a bias b, which construct the hyperplane that separates the two data classes (Figure 3-4a). The decision function in the data space is:

$$f(x) = sign(w^T x + b) \tag{3.4}$$

The optimization of equation 3.3 gives us a maximal margin classifier. However, this method, which is called "Hard margin", only solve the problem when two classes are separated. In the case when two classes are not separable, all constrains can not be satisfied. This property makes this method susceptible to noise, where an extreme value (noise) can make a huge impact on the decision function. To overcome this problem, the l_1 regularization is applied to the optimization function. A slack variable ξ_i is added to the function to penalize the outliers. The "Soft margin" classification problem become:

$$\begin{aligned} min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i(w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, ..., n \end{aligned} \tag{3.5}$$

By solving this optimization problem, we can obtain a linear boundary as in Figure 3-4b where some of the misclassified samples are tolerated. The Lagrangian duality problem is often used to solve this problem. After applying the Lagrangian coefficient



(a) Hard margin SVM (b) Soft margin SVM

Figure 3-4 – Support Vector Machine

α, the optimization in equation 3.5 becomes:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \\ s.t. \quad & 0 \leq \alpha_i \leq C, \quad i = 1, ..., n \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{3.6}$$

with the decision function:

$$f(x) = sgn(\sum_{i=1}^{n} y_i \alpha_i \langle x_i, x \rangle + b) \tag{3.7}$$

The non-linear classification problem could be solved by replacing the dot product $\langle x_i, x_j \rangle$ by a mapping function ϕ: $\langle x_i, x_j \rangle \rightarrow \langle \phi(x_i), \phi(x_j) \rangle$ to map the dataset vector space into a high dimensional feature space $\mathbb{H}$ where two classes can be separated more easily. However, in many cases the function ϕ could not be defined explicitly. So thanks to RKSH theory we can choose a kernel that defines a dot in feature space as $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Kernels such as Radial Basis Function (RBF) or polynomial kernels are frequently used. The details of the kernels will be described later in section 3.2.4.

- **One-class Support Vector Machine**

  In the real world, the amount of unlabeled data is much larger than the labeled data, and the labeling task could be costly and takes much effort. So the unsupervised one-class SVM technique [94] is developed to take advantage of these unlabeled dataset. In one-class SVM, we assume that most of the samples in the data space $\mathbb{X}$ follow an unknown probability P [29] and the objective is to find out the region that contains these samples. The solution is to find a hyperplane in a feature space $\mathbb{H}$, which maximize the distance of the training data from the origin in this space, while only a small fraction of the data falls between the hyperplane and the origin. The corresponding optimization problem can be written as:

$$\begin{aligned} min_{w,b} \quad & \frac{1}{2}\langle w, w\rangle + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \\ s.t. \quad & \langle w, \phi(x_i)\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in [1...n] \end{aligned} \tag{3.8}$$

  where $x_i \in \mathbb{X}$, are training data in space $\mathbb{X}$, $\phi : \mathbb{X} \to \mathbb{H}$ is a mapping function that maps the data from normal space $\mathbb{X}$ to $\mathbb{H}$, the function $\langle w, \phi(x_i)\rangle - \rho = 0$ is the decision hyperplane, ξ_i is slack variables, $\nu \in (0, 1]$ is the hyperparameter that control the impact of constrains violation and the overall cost. The one-class SVM model in feature space is sketched in Figure 3-5.

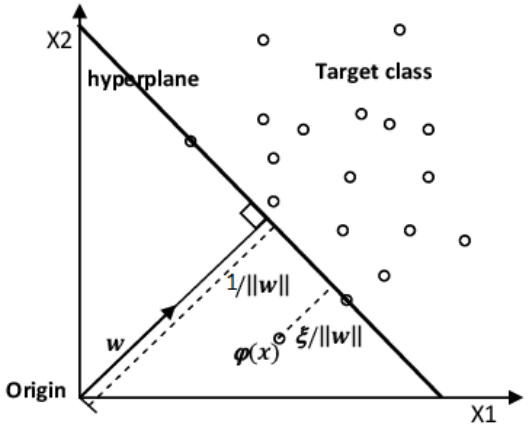


Figure 3-5 – One class SVM in feature space.

In the SVM model, the mapping function ϕ is used when two classes are not separable by a linear decision function. On the contrary, the one-class SVM model can not work

without this function. The Lagrangian duality for the one-class SVM's optimization problem is constructed:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha_i\alpha_j\langle\phi(x_i),\phi(x_j)\rangle \\ s.t. \quad & 0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad \sum_{i=1}^{n}\alpha_i = 1 \quad \forall i \in [1...n] \end{aligned} \tag{3.9}$$

The decision function is defined as:

$$f(x) = sgn(\sum_{i=1}^{n}\alpha_i\langle\phi(x_i),\phi(x)\rangle - \rho) \tag{3.10}$$

The one-class SVM technique is mainly used for anomaly detection ([42], [72], [107], [101]) or novelty detection ([109], [125], [26]). It is one of the most popular technique in machine learning for this task due to its effectiveness in non-linear cases.

3.2.3 Dissimilarity measure

In the detection anomaly problem, one of the most crucial tasks is comparing two instances in the dataset. Many techniques rely on the ability to measure dissimilarity or distance between data instances to determine whether an instance belongs to the normal or abnormal class. Two data types of interest in the context of this work are point in vectorial space and sequence. The dissimilarity measure affects deeply the detection result. In this section, we present the conditions of a dissimilarity measure, then we introduce distance measures for point data, and next comes the distance measures for sequence. Some popular dissimilarity measure has been reviewed in [129]. Weller and al. [126] have presented a review of the use of dissimilarity and distance measures within network intrusion anomaly detection (NIAD), among those distances have stronger mathematical ground and are chosen most of the time. Let first define the characteristics of dissimilarity measures φ:

1. Non-negativity: the distance between two instances A and B is always greater or equal zero.

$$\varphi(A,B) \geq 0 \tag{3.11}$$

2. Identity of indiscernible: the distance between A and B is equal to 0 if and only if A equal to B.

$$\varphi(A, B) = 0 \quad \text{iff} \quad A = B \tag{3.12}$$

3. Symmetry: the distance between A and B equals to the distance between B and A.

$$\varphi(A, B) = \varphi(B, A) \tag{3.13}$$

4. Triangle inequality: the distance between A and B is always less or equal to the sum of the distance between A and C and the distance between B and C.

$$\varphi(A, B) \leq \varphi(A, C) + \varphi(B, C) \tag{3.14}$$

In the context of security monitoring, we will deal mainly with vectorial data and sequence. So we review the most popular distances for those two types of data.

Dissimilarity measure for point

Suppose that we have two vectors SX and SY with the same length N, where $SX = [sx_1, sx_2, ..., sx_N]$ and $SY = [sy_1, sy_2, ..., sy_N]$, these vectors could be considered as two points in a multi dimensional space. According to a survey in [123] and [24], there are 8 families of dissimilarity measure. The most popular is the Euclidean Distance (ED), a.k.a l_2-norm and the Manhattan distance (l_1-norm) distance in the Minkowski family (equation 3.15). These similarity measures have been applied in most of machine learning task.

$$\varphi_{Minkowski}(SX, SY) = \sqrt[r]{\sum_{i=1}^{N}(|sx_i - sy_i|^p)} \tag{3.15}$$

However, they do not support the comparison between two sequences with different lengths. So sequence-based similarity measure are proposed to solve this problem.

Dissimilarity measure for sequence

The second part of similarity measure discusses the methods to calculate the similarity between sequences. Suppose that we have two sequences SX and SY with length N and M respectively. So $SX = (sx_1, sx_2, ..., sx_i, ..., sx_N)$ and $SY = (sy_1, sy_2, ..., sy_j, ..., sy_M)$. One of the most popular algorithm to compare two sequences is the Dynamic Time Warping (DTW) method. This method was proposed to allow some points to repeat as many times as needed to get the best alignment. The output of this method is a value that describes the dissimilarity or distance between two sequences, so one could call it DTW distance. DTW was first introduced in [99], [100] for the speech recognition problem. But it is also applied in a wide range of application, such as classification ([89], [50]) or clustering ([59], [57]). The basic idea of DTW is to calculate the warping path between two sequences that give the minimum cumulated distance between them.

An alignment path π of length L between SX and SY is a sequence of points $\pi = (\pi(1), \pi(2), ..., \pi(l), ..., \pi(L)$, with $\pi(l) = (\pi_1(i), \pi_2(j)), i \in [1 : N], j \in [1 : M]$, $l \in [1 : L]$. Meaning that the i^{th} element of sequence SX is going the match the j^{th} element of sequence SY. The alignment path (warping path) between two sequences defines the corresponding element in one sequence to another element in the other sequence, which assign first and last elements of two sequences to each other [106]. The warping path must satisfy the following criteria:

1. Boundary condition: $\pi(1) = (1, 1)$ and $\pi(L) = (N, M)$. The starting point and ending point of the path is the first and last point of the two sequences.

2. Monotonicity condition: the wraping path does not go inverse in either sequence.

3. Step size condition: the step is limited by a condition: $\pi(l+1)-\pi(l) \in [(1, 1), (1, 0), (0, 1)]$.

This algorithm starts by computing a cost matrix C between every element of two sequences SX and SY.

$$C(i, j) = \varphi(sx_i, sy_j) \tag{3.16}$$

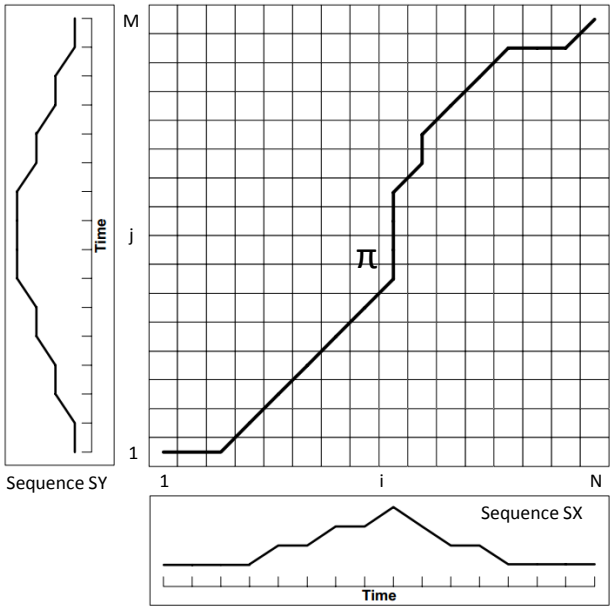


Figure 3-6 – Dynamic time warping alignment.

In case each element of a sequence is a numeric vector, we can apply a point-based similarity measure φ such as l_2-norm. In some special cases, where each element of the sequence contains symbols or character, we have to define a custom similarity measure appropriate for the context.

Next, the accumulated cost function D_π associated to a warping path π is computed in equation 3.17, which calculates the total cost of an alignment path π between two sequences SX, and SY. The two sequences may have different lengths, but they are aligned with a path π as in Figure 3-6. There are many alignment paths between two sequences, which could be understood as many ways to connect two points $(1, 1)$ and (N, M).

$$D_\pi(SX, SY) = \sum_{l=1}^{L} \varphi(sx_{\pi_1(l)}, sy_{\pi_2(l)}) \tag{3.17}$$

Finally, we calculate the DTW distance between two sequences SX and SY (equation 3.18), which is the minimum cost of the cumulated cost matrix D associated to an optimal

warping path $\pi_{optimal}$:

$$DTW(SX, SY) = D_{\pi_{optimal}}(SX, SY) = \min_{\pi}(D_{\pi}(SX, SY), \pi \in P^{N \times M}) \tag{3.18}$$

where $P^{N \times M}$ is the set of all possible alignment paths between SX and SY. The optimal path is determined by using dynamic programming, with the objective to find the optimal sub-total cost at each point in the process. The dynamic program is expressed:

1. First row: $D(1, j) = \sum_{k=1}^{j} C(1, k)$.
2. First column $D(i, 1) = \sum_{k=1}^{i} C(k, 1)$.
3. Other elements: $D(i, j) = min(D(n-1, m-1), D(n-1, m), D(n, m-1)) + C(i, j), i \in [1, N], j \in [1, M]$.

The time complexity of this equation is $O(N, M)$. In other words, the DTW distance between sequence SX and SY is computed at $D(M, N)$. By introducing the DTW algorithm, we have presented one of the most popular technique to compare two sequences. The anomaly detection methods presented in section 3.2.2 and section 3.2.1 usually take point data as input. But now one can apply these methods for sequence data if we compute the distance between sequences by using DTW method.

Section 3.2.3 introduced dissimilarity measures for point data and for sequence data. Beside dissimilarity measure, kernel method is also important for the anomaly detection methods such as SVM or one-class SVM. Section 3.2.4 will first explain about the role of kernel method, and its implementation in SVM and one-class SVM. Then two categories of kernels, one for point data and one for sequence data are presented.

3.2.4 Kernel method

Kernel method has become a popular tool for anomaly detection techniques [105]. Normally, it is associated with the SVM method and the one-class SVM method to solve

the non-linear classification problem [33]. However, kernels can also be used for clustering [39], and for outlier detection [105]. The key insight of the kernel method is that many machine learning techniques like SVM or K-means use the dot-product between between vectors when creating a model of the data, and that this dot-product can be replaced by a mathematically more general notion called a kernel. Many types of kernels create an implicit mapping of objects into a higher dimensional feature space as mentioned in section 3.2.2. This mapping is implicit, because the kernel function directly computes the dot-product in this higher dimensional feature space, without computing the mapping. Computing this is often much cheaper than doing the mapping explicitly, if it all, and computing the dot-product in the high dimensional space. In the section 3.2.2, a mapping function ϕ was defined to map data points from normal data space to a higher dimensional feature space. A kernel is defined as:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{3.19}$$

where the inner product $\langle x_i, x_j \rangle$ in normal space becomes $\langle \phi(x_i), \phi(x_j) \rangle$ in feature space, and we could replace $\langle \phi(x_i), \phi(x_j) \rangle$ with the kernel $K(x_i, x_j)$. The kernel matrix is also called a Gram matrix. So as we apply the kernel to SVM and one-class SVM model, we can train these models in the high dimensional feature space given by ϕ, without having to explicitly calculate $\phi(x)$. With the kernel K, the optimization of SVM in equation 3.6 becomes:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ s.t. \quad & 0 \leq \alpha_i \leq C, \quad i \in [1:n] \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{3.20}$$

and the decision function is: $f(x) = sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b)$. Similarly, the

optimization problem of one-class SVM in equation 3.9 becomes:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha_i\alpha_j K(x_i, x_j) \\ s.t. \quad & 0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad i \in [1:n] \\ & \sum_{i=1}^{n} \alpha_i = 1 \end{aligned} \tag{3.21}$$

The decision function is defined as:

$$f(x) = sgn(\sum_{i=1}^{n} \alpha_i K(x_i, x) - \rho) \tag{3.22}$$

A kernel needs to satisfy Mercer conditions ([8], [98]).

Definition 3.2.1. *Let $X = x_1, x_2, ..., x_n$ be a nonempty set where $x_i \in R^d$. A function $K : X \times X \to R$ is called a positive definite kernel (Mercer kernel) if and only if K is symmetric and the following equation holds:*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K(x_i, x_j) \geq 0, \quad a \in R^n, \quad a \neq 0 \tag{3.23}$$

Note that a model that implement a not positive definite kernel can also give a good detection result in practice, this has been proved by Gudmundsson et al. [54]. Many types of kernel were developed, but we can categorize them into two groups corresponding to point data and sequence data.

Point kernel

There are some examples of Mercer kernels that has been introduced for a dataset that contains an ensemble of points $X = [x_1, x_2, ..., x_n]$, $x_i \in R^d$:

- Linear kernel: $K_{linear}(x_i, x_j) = x_i \cdot x_j$
- Polynomial kernel of degree p: $K_{poly}(x_i, x_j) = (1 + x_i \cdot x_j)^p$, $p \in N$

- RBF kernel: $K_{rbf}(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, $\sigma \in R$.

- Tanh kernel: $K_{tanh}(x_i, x_j) = (\beta x_i \cdot x_j + b)$. The tanh kernel only satisfies Mercer's condition for certain values of β and b.

These kernels are popular for point data, and they are applied for many application such as clustering ([53], [92]) or classification [23].

Sequence kernel

We consider three types of kernels that can be applied to sequences, each of them has its advantages, and disadvantages. To explain the kernels in this section, suppose that we have a set of K sequences, SX and SY are two representing sequences in this set with the length N and M respectively, where $SX = [sx_1, sx_2, ..., sx_N]$ and $SY = [sy_1, sy_2, ..., sy_M]$, sx_i and sy_j have similar features. First, we define the elementary kernel k which could be a basic kernel defined above such as RBF kernel or polynomial kernel.

The first kernel is the Global Dynamic Time Warping (GDTW) kernel. This kernel has been used in many application for time series data ([4], [130], [54]). The equation 3.24 computes the Gram matrix of the GDTW kernel.

$$K_{GDTW}(SX, SY) = \exp(-\frac{1}{\sigma^2} DTW(SX, SY)^2) \tag{3.24}$$

The GDTW kernel uses the DTW similarity measure between two sequences and then apply the RBF kernel for the output of DTW algorithm. This kernel has not been proved to satisfy the positive definite characteristic, but its performance in some applications is considerable [54].

The second kernel for sequence is the Dynamic Time Alignment (DTA) kernel, which was introduced in [10] and applied for the SVM classification problem [111] and detect facial gesture [133]. This kernel is also used for temporal data ([115], [7], [97]). DTA kernel considers the arithmetic mean of the kernel values along the alignment path. This kernel is

developed with the idea of using non-linear mapping function for each sequence, where

$$\Phi(SX) = (\phi(sx_1), \phi(sx_2), ..., \phi(sx_N)) \tag{3.25}$$

The DTA kernel is introduced in equation:

$$\begin{aligned} K_{DTA}(SX, SY) &= \langle \Phi(SX), \Phi(SY) \rangle \\ &= \max_{\pi_1,\pi_2} \frac{1}{M_{\pi_1,\pi_2}} \sum_{l=1}^{L} m(l) \langle \phi(sx_{\pi_1(l)}) \phi(sy_{\pi_2(l)}) \rangle \\ &= \max_{\pi_1,\pi_2} \frac{1}{M_{\pi_1,\pi_2}} \sum_{l=1}^{L} m(l) k(sx_{\pi_1(l)}, sy_{\pi_2(l)}) \end{aligned} \tag{3.26}$$

where π_1, π_2 are alignment path of length L of the optimal path between SX and SY. $m(l)$ is a nonnegative weighting coefficient and $M_{\pi_1,\pi_2} = \sum_{l=1}^{L} m(l)$ is a normalization factor.

In equation 3.25, the kernel $K_{DTA}(SX, SY)$ between two sequences is calculated by breaking the sequence-mapping function Φ into smaller element-mapping function ϕ. Then a basic kernel k is applied to replace the mapping function ϕ of the optimal alignment path and the mean of kernel k along the path is used as the kernel. This kernel was proved to be symmetric and satisfy the Cauchy-Schwartz inequality [111], but the positive definite property of this kernel is not proved either.

Note that the optimal internal equation is changed to max function instead of original min function of DTW. While the similarity used for GDTW kernel is a similarity measure, such as l_2-norm, the similarity measure for DTA kernel is a kernel such as RBF. This kernel is quite similar to the GDTW, but the implementing order between the basic kernel k and the DTW algorithm is inverted.

The Global Alignment (GA) kernel, which was presented by Cuturi et al ([36], [35]) emphasized the fact that these two previous kernels have not been proven to be positive definite. The GA kernel considers the values obtained with all the possible alignments and it was proved to satisfy the positive definite condition. This kernel has been used to solve problems such as classification ([64], [63]) or regression [88].

To explain the GA kernel we first recall the definition of alignment path π_1 and π_2

and the cumulation cost matrix of DTW algorithm in equation 3.17. So we rewrite the equation 3.17:

$$D_\pi(SX, SY) = \sum_{i=1}^{|\pi|} \varphi(sx_{\pi_1(i)}, sy_{\pi_2(i)}) \tag{3.27}$$

where $|\pi|$ signifies the length of the path π. φ can be defined as ED distance: $\varphi(sx_i, sy_j) = \|sx_i - sy_j\|^2$. The GA kernel is defined as:

$$K_{GA}(SX, SY) = \sum_{\pi \in A(N,M)} e^{-D_\pi(SX,SY)} \tag{3.28}$$

where $A(N, M)$ is the set of all alignments between two sequences SX and SY of length N and M respectively. By defining a function $\kappa = e^{-\varphi}$, equation 3.28 becomes:

$$K_{GA}(SX, SY) = \sum_{\pi \in A} \prod_{i=1}^{|\pi|} \kappa(sx_{\pi_1(i)}, sy_{\pi_2(i)}) \tag{3.29}$$

The following results are also proved [35]:

1. K_{GA} is positive definite if κ is such that $\frac{\kappa}{1+\kappa}$ is p.d.
2. K_{GA} is likely to yield diagonally dominant Gram matrices when used on real datasets.
3. The computation required to compute K_{GA} is $O(MN)$, similar to the DTW distance. The value $K_{GA}(SX, SY)$ in equation 3.28 is equal to $G(N, M)$ where the matrix G is calculated as:

$$G(i,j) = \begin{cases} 1 & \textit{if i=0 and j=0} \\ 0 & \textit{if i=0 or j=0}, i \neq j \\ \kappa(sx_i, sy_j)(G(i-1,j-1) + G(i-1,j) + G(i,j-1)) & \textit{otherwise} \end{cases} \tag{3.30}$$

.

The similarity measure induced by this kernel is different from the previous alignment

kernels. Indeed, the sum in equation 3.28 takes advantage of every possible alignment instead of only the optimal one. Thus, two sequences are similar in the sense of K_{GA} not only if they have an alignment which results in a small DTW distance, but also share numerous suitable alignments. In practice, equation 3.29 is usually replaced by its logarithm. The GA kernel is thus defined as:

$$K_{GA}(SX, SY) = \log(\sum_{\pi \in A} \prod_{i=1}^{|\pi|} \kappa(sx_{\pi_1(i)}, sy_{\pi_2(i)})) \tag{3.31}$$

The log version of the GA kernel does not hold the positive definite constraint, but it is used in many researches ([64], [35]), which have achieved good classification performances. Since many kernels were proposed, it is difficult to compare the performance of each kernel. So the Kernel Alignment (KA) method [34] is proposed by Cristianini et al. as a measure of similarity between two kernel functions or between a kernel and a target function. This method can also be used as an evaluation metric for kernel.

Kernel alignment

The kernel alignment was presented in [34] with the objective to calculate the similarity between Gram matrices. Given a set S with L elements: $S = \{x_i, y_i\}_{i=1}^{L}$, where x_i is data instance (vector or sequence) in input space and $y_i \in \{-1, +1\}$ are labels. Each instance x_i is mapped to a feature space by a mapping function ϕ and the kernel are used to replace this function. Suppose that we use two different Gram matrices K_1 and K_2 (different type of kernel or same type but different hyperparameters), the kernel alignment method is defined as:

$$A(S, K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \tag{3.32}$$

where $\langle K_1, K_2 \rangle_F$ is the Frobenius inner product between two matrices. The equation 3.32 provides the similarity between two kernels. For arbitrary matrices, this score range between -1 and 1, the higher value of $A(S, K_1, K_2)$, the more similar. However, since kernel

matrices are semi-definite positive, the score is lower-bounded by 0 [108].

Another way to implement this method is to define the ideal kernel matrix G as $G = YY^T$, where $Y = [y_1, ..., y_L]$ is the label vector. this matrix proposes a clear separation between normal and abnormal classes. So other kernels are compared to the ideal kernel G instead of comparing to each other, and the kernel most similar to G is supposed to have best performance. The equation 3.32 becomes:

$$A(S, K, YY^T) = \frac{\langle K, YY^T \rangle_F}{\sqrt{\langle K, K \rangle_F \langle YY^T, YY^T \rangle_F}} = \frac{\langle K, YY^T \rangle_F}{m\sqrt{\langle K, K \rangle_F}} \tag{3.33}$$

where K is a Gram matrix of a kernel and $\langle YY^T, YY^T \rangle_F = m^2$. Igel et al. [58] proposed another way to express the kernel alignment. Formally, maximizing $A(S, K, YY^T)$ is equivalent to minimizing the following distance:

$$d(K, YY^T) = \sqrt{2 - 2A(S, K, YY^T)} \tag{3.34}$$

Based on this similarity we can search for the kernel that gives the best performance for the detection. Wang et al. [124] discussed the application of KA methods such as tuning kernel hyper parameters [22], clustering [71] and approaches using multiple kernels ([75], [113]).

So until this point, we discussed the anomaly detection techniques for both point and sequence data. The next part of this chapter presents the anomaly detection issues related to trajectory data, which is a type of sequence. It gives an overview on how other researchers process this kind of data in some applications.

3.3 Issue related to trajectory data

This section discusses the anomaly detection approaches for trajectory data. The definition of trajectory data and its source are presented first. Then the approaches that detect abnormal behaviors using trajectory data are discussed. These approaches could

be categorized into two groups [131]. The first group analyze multiple trajectories and extract key features, abnormal state in a region could be detected using these features. The second group classifies individual trajectory, where a classifier is built to detect abnormal trajectory. The approaches in these groups will be presented respectively, although the purposes of these works are not limited to security issues.

3.3.1 Trajectory

The development of new technology has created a massive number of location-aware devices in recent years, which led to an enormous number of trajectory data obtained. The sources of trajectory data is very diverse: global positioning system (GPS) device ([118], [87]), global system for mobile communications (GSM), radio frequency identification (RFID) [134], wifi [112] and camera ([110], [21]). GPS based data is composed of temporally ordered sequences of geographic coordinates recorded by a GPS-enabled device carried by the moving object such as GPS in cars, mobile phone. GSM based data is composed of temporally ordered sequences of identifiers of the cells in which the moving object passes. RFID based data contain a sequence of identifiers of RFID readers through which the moving object passed such as the luggage transfer system at the airport, where the RFID on the luggage is controlled [5]. Wi-Fi-based data contain a sequence of identifiers of access points that communicated with the moving object. The camera trajectories come from surveillance camera setup in public space, such as on highway [76], shopping center [19] or office [78]. Despite a diverse source of trajectory, in general this type of data is spacial-temporal data. The definition of trajectory is as follow:

Definition 3.3.1. *A trajectory is formally represented as a vector T of movement points: $T = [p_1, p_2, ..., p_n]$, where $p_k = (ID, t_k, pos_k)$ is the k^{th} element of the trajectory T, ID is the identity of the trajectory's subject, t_k is the time at which the point was recorded, and pos_k is the coordinate of this point.*

With the definition of trajectory, many works has been done for the anomaly detection task using this type of data. The next part presents the first approach to analyze trajectories and detect abnormal event in an area.

3.3.2 Identifying anomalous events using trajectories

The first category detects anomalies in an area by using many sub-trajectories in a time frame. The anomalies in this category are caused by the presence of one or many abnormal trajectories combined. Researchers tend to divide the whole environment into multiple small areas, then only the sub-trajectory that lie inside the critical area is considered.

Liu et al. [74] partition a city into multiple main regions connected by major roads, and try to detect abnormal behaviors on the links between regions according to the trajectories of vehicles traveling on those links. They divide a day into time bins and identify for each link three features: the number of vehicles traveling a link in a time bin, proportion of these vehicles among all vehicles entering the destination region, and vehicles departing from the origin region. The features of a time bin were respectively compared with those in the equivalent time bins of previous days to calculate the minimum 'distort' of each feature. In other words, to find the similar patterns between different days. Later, the Mahalanobis distance is used to measure the extreme points, which are regarded as outliers.

Pang et al. ([84] [85]) adapt likelihood ratio test, which has previously been used in epidemiological studies, to describe traffic patterns. They use uniform grids to partition a city, and count the number of vehicles arriving in a grid cell over a period. The objective is to identify the contiguous set of cells and time intervals which have the largest statistically significant departure from expected behavior (i.e., the number of vehicles). The regions whose log-likelihood ratio statistic value drops in the tail of χ^2 distribution are likely to be anomalous [25].

The PROSAFE project [16] developed a smart home surveillance system to detect abnormal behavior to help elderly people and patient. Trajectories of people inside a hospital are captured by infra sensors and features such as agitation, immobility, were used for analysis. The optimal time division in a day is computed using the genetic algorithm. A threshold for each feature in each period is defined to detect abnormal behavior of the patient to raise the alarm.

The similarities of these works are, first, the environment (city, house) is divided into areas, and the key metrics (feature in time bins) are extracted from the trajectories

in each area. Second, their objective is to detect abnormal behaviors, not the abnormal trajectories. These key metrics are kept to train classifiers in order to later detect abnormal events. Our first method AED to detect abnormal event was motivated by this direction. It concentrates on critical areas in the building and therefore could detect abnormal behaviors of users in these locations easily. More detail about this method will be given in chapter 4.

3.3.3 Detecting outlier trajectories

The second approach for anomaly detection using trajectory data is analyzing the whole trajectory. The original trajectory (definition 3.3.1) is efficient for short passages like a street intersection or a public zone ([6], [68]). This kind of trajectory is sensitive because the sequence of precise coordinates of everyone's movement is different, so it's very difficult to classify an abnormal trajectory from normal ones, and models that analyze this kind of trajectory could face a high false alarm rate in detection task. Moreover, the time and duration of the trajectory is not considered in many approaches. With these disadvantages of analyzing sequence of movement points, the first issue is to choose the appropriate form of data for the context.

The graph representation of trajectory is one of the most popular methods to represent trajectory, where the environment (city, building) is divided into multiple sub-areas (nodes) and connections (edges). Many researchers have used the terms Point of Interest (POI) and Activity Path (AP) to represent the nodes and edges respectively. And trajectories are mapped into sequence of POI and AP. This kind of sequence was applied in the location travel recommendation application. Zheng et al. [132] proposed to find the interesting locations and travel sequences from trajectories generated by many people. In the method, authors first detect stay points (POI) from each trajectory and then cluster the stay points of different people into locations. Based on these locations and raw trajectories, they build a user-location graph to decide which location is suitable for which user.

Boukhechba et al [17] constructed sequence of POIs that represents the tracking of user's daily habits. Every sequence is stored incrementally in a tree structure called Habits Tree. Finally, an algorithm is trained to predict the next POI using the association rules drawn from the Habit Tree.

In ([79],[78]), POI is the entry/exit of the surveillance zone or the zone where the tracked person stays longer than a threshold, and AP is the moving path between POI. Authors presented a general framework for live video analysis. The POI are learned by clustering trajectories, then authors represented a graph by a hidden Markov model to perform multiple tasks such as classification, future behavior prediction, and anomaly detection.

Pan et al. [83] analyze data from GPS of taxis in Beijing. The road network is mapped into a graph where nodes represent the terminal points of road segments and edges are road segments. Authors try to find abnormal drivers' routing behaviors to monitor all the traffic anomalies with accurate diagnoses of their impact regions - regions where anomalies happen, and relevant descriptive terms. Other kinds of trajectory representation are out of the scope of this thesis.

By presenting these works, we realize that graph representation is an effective solution to represent trajectories. This representation map trajectories into sequence of nodes and edges, which can reduce the size of trajectories and give more information. Therefore, our second method DTA were created using this graph representation as the key insight. Its process will be explained in chapter 5.

3.4 Conclusion

In this chapter, we discussed the anomaly detection problem for trajectory data. In the first part, the general definition of anomalies and main approaches to the detection of anomaly which use the density and the boundary of the data are presented. The second part of this chapter concentrates on the applications and the approaches related to trajectory analysis.

By studying the way other researchers process the trajectory data, we have identified two main approaches. The first one is detecting abnormal events using key metrics based on trajectories, and the second approach is detecting abnormal trajectories using trajectories directly. Based on this point of view, we proposed two methods: *Abnormal event detection using key metrics* and *Abnormal trajectories detection*, to detect anomalies in the critical

infrastructures. The first method is a parametric method, which process trajectories as events inside important sub-areas in the building. Chapter 4 will discuss this approach. The second method (ATD) in chapter 5 deals with sequence of activities extracted from trajectories. Both of the methods are applied to both scenarios (commercial building and shopping center). Due to the different characteristics of these scenarios, each technique is adapted to suit the specific context.

Chapter 4

Abnormal events detection using key metrics

4.1 Introduction

This chapter presents the *Abnormal events detection using key metrics* method, the first proposed method for VIRTUALIS project. This technique is motivated by an idea of constructing a detection method to protect a local area inside the building, which is inspired by approaches described in Section 3.3.2. After the methodology introduction, we go further in the explanation of how we selected the critical areas, the key metrics, how to compute them, and how to train the classifiers. Then we discuss the implementation of the two scenarios: the working office and the shopping center given by the VIRTUALIS and ATC datasets respectively.

4.2 Methodology

This section presents the methodology of this technique and the definition of some essential elements. The proposed detection method aims to apply for critical and vulnerable local areas in a building.

4.2.1 Training and detection stage

This method is composed of three stages: training, validation, and detection stage.

In the training stage, the first step is to pre-define the parameters of time windows and select the critical areas. At each time window, we collect all the points of a trajectory inside the chosen area (sub-trajectory). These sub-trajectories are also called events, which could be understood as the presence of people in an area. Events in the same time window are used to compute the key metrics, which are features characterizing people's activities that can be used to detect abnormal behavior. The types of key metrics used in the experiments are for example *average duration of a visit*, *number of visits*. The vector that contains the key metrics of the same time window is called vector of key metrics. Then a classifier is trained using the vectors of key metrics of multiple days in the dataset. There are two proposed classifiers: the ensemble classifier and the one-class SVM classifier.

The validation stage tests the trained classifier on offline mode. The validation set is a labeled dataset, which includes events describing regular situations and attack scenarios. This step allows to calibrate the parameters of the classifier. The evaluation metrics that are used are true positive rate, and false negative rate. The accuracy is not used because only a few abnormal samples are available in the training set.

The detection stage is done online, where a classifier is deployed to detect abnormal events in real time.

In this manuscript, we present the results of the training and validation stages. The training stage is composed of the following steps:

1. Define essential elements such as critical areas, time window's parameters or the key metrics.

2. Assign the raw sub-trajectories in the critical area from the training set to each time window.

3. Compute the key metrics using the sub-trajectories inside each time window to create vectors of key metrics.

4. Train one or several classifiers using the vectors of key metrics.

The detection stage is based on the following principle: the trajectories are observed in real time, new vectors of key metrics are calculated, and these vectors are classified using the trained classifiers. The detection process is as follow:

1. Continuously collect the trajectory from cameras and sensors .
2. Select only the sub-trajectories in the predefined zone.
3. Compute the vector of key metrics in the current time window.
4. Predict the label of a new vector of key metrics using the classifier.
5. If the observed value is detected as abnormal, an alarm is set.

The validation stage is similar to the detection stage, except it happens offline. The experiments in this chapter include the training and validation stages.

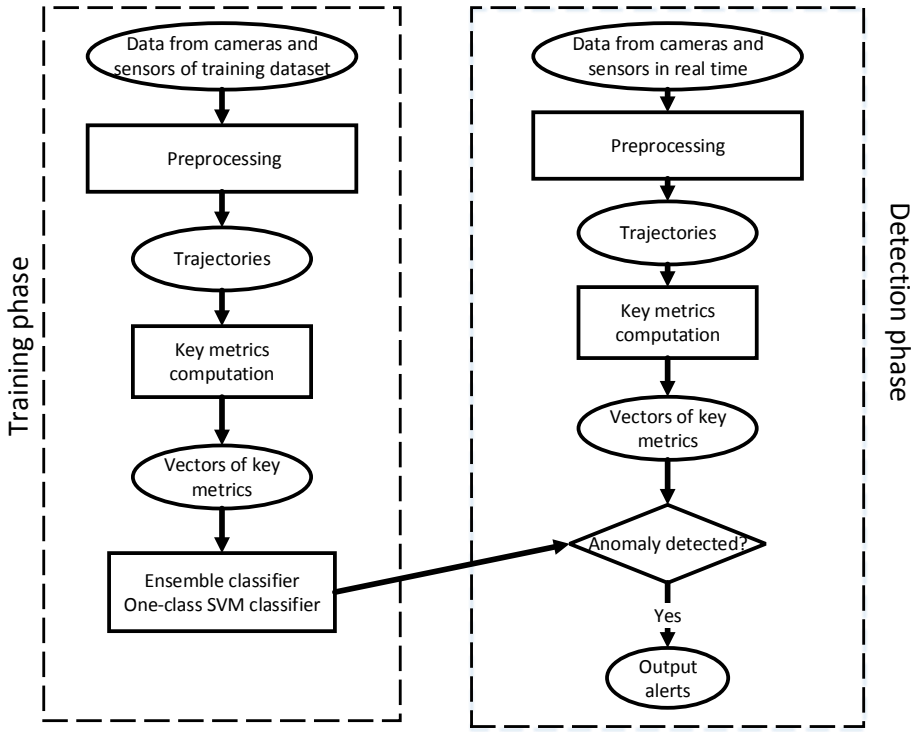


Figure 4-1 – Training and detection stages of the AED method.

This method detects the abnormal key metrics in a specific time window. However, in most of the situation, it does not point out the individual event which causes the abnormal key metric because a key metric in a time window is the result of multiple events. The following part explains further the terms appeared in the methodology such as critical area selection, time window, key metrics, and classifiers and their affections in the proposed method.

4.2.2 General description

Critical area selection

This method requires to define some critical areas of the environment. These areas can be obtained by dividing the place using its structure, for example, each room inside a building is an area. In a building, a critical area can be defined as the area around a vital element (an element which is essential and attracts people) such as a printer, an electrical panel, a ticket machine in a railway station, etc. Another possibility consists in using the specific characteristics of the pedestrians such as areas where people move and areas where people stay. The zone segmentation allows to define a list of areas. Since only the critical areas are chosen to apply the method, a selection is based on the vulnerability and criticality of the area.

Time window selection

Key metrics are features that describe some characteristics of an area in a period. The time window is used to separate a day into multiple time frames, and it has two hyperparameters: the duration and the shift. Time windows overlap when the shift is smaller than the duration. The hyperparameters are chosen by experience. If the duration is too large, all events that happen in a window are merged, so the time factor is partly lost. On the contrary, if the duration is too short, the number of observed data in the windows may be too small to obtain a reliable estimation of the key metrics. Besides, it has to be noted for the sliding window that the computation may be expensive if the shift is small.

Key metric vector

In this technique, we are interested in examining the behavior of people concerning their presence in a critical area, so we define key metrics to describe the situation of an area in a time window. The values of key metrics are organized into vectors of key metrics determined as below:

Notation 4.2.1. *The vector of key metrics X_t, $X_t \in R^{K \times D}$ associated with time window t is a two dimensional vector, where each row corresponds to a specific day in the dataset and each column corresponds to a particular key metric. Each element in this vector is $m^k_{t,d}$, corresponding to an observed value of the key metric k, at time window t at day d, $k \in [1:K]$, $d \in [1:D]$ where D is the number of day and K is the number of key metric.*

Table 4.1 – Example of vector of key metrics at time window t.

$m^k_{t,d}$	$k=1$	$k=2$	$k=3$
$d=1$	$m^1_{t,1}$	$m^2_{t,1}$	$m^3_{t,1}$
$d=2$	$m^1_{t,2}$	$m^2_{t,2}$	$m^3_{t,2}$
$d=3$	$m^1_{t,3}$	$m^2_{t,3}$	$m^3_{t,3}$
$d=4$	$m^1_{t,4}$	$m^2_{t,4}$	$m^3_{t,4}$
$d=5$	$m^1_{t,5}$	$m^2_{t,5}$	$m^3_{t,5}$

An example of vector of key metrics is showed in Table 4.1, with $D = 5$ and $K = 3$.

This notation will be used throughout this chapter to create the type of data specified for two types of classifier: ensemble classifier and one-class SVM classifier. The key metrics selection is crucial because it defines the feature to classify a behavior as abnormal or normal. The primary types of key metrics that we propose are *average duration of stay*, *maximum duration of stay*, *number of visits* and *occupation rate*, which can be varied in three fields: type of people, time and location. It means with these types of key metrics, specific key metrics are given for different types of people, time periods or locations according to the input data. For example, the key metric *average duration of each visit of an engineer at the printer zone* is created by computing a basic key metric (*average duration*) for a specific type of employee (engineer) at a particular location (printer zone).

The computation of the value $m^k_{t,d}$ is different depends on the key metrics. For example, with the key metric *average duration of a visit*, this term is calculated by taking

the mean of every event's duration with a condition that those events started in the same time window. And with the key metric *number of visits*, this term is obtained by counting number of events happened in a time window.

Ensemble classifier

This classifier creates an alert for each key metric. This classifier is built by computing a threshold for each key metric in each time window.

Notation 4.2.2. *The random variable M_t^k is a key metric that has one value in each day: $M_t^k = [m_{t,1}^k, m_{t,2}^k, ..., m_{t,d}^k, ..., m_{t,D}^k]$, where K is number of key metrics and D is number of day in the dataset, $k \in [1:K]$, $d \in [1:D]$.*

The data used in this classifier is the realizations of the variable M_t^k. And the following assumption is made:

Assumption 4.2.1. *The key metric M_t^k associated to a time window t is a random variable which is assumed to fit a normal distribution with mean $\mu_{M_t^k}$ and standard deviation $\sigma_{M_t^k}$: $M_t^k \sim N(\mu_{M_t^k}, \sigma^2_{M_t^k})$.*

This assumption is described in Figure 4-2, where at each time window, M_t^k fits a normal distribution in the vertical axis which represents observed values in multiple days.

In the training stage, the mean $\mu_{M_t^k}$ and the standard deviation $\sigma_{M_t^k}$ of the key metric M_t^k are estimated using the data in the training set composed of D days, assumed to contain only normal events, according to:

$$\hat{\mu_{M_t^k}} = \frac{\sum_{d=1}^{D} m_{t,d}^k}{D} \tag{4.1}$$

$$\hat{\sigma_{M_t^k}} = \sqrt{\frac{1}{D-1}\sum_{d=1}^{D}(m_{t,d}^k - \hat{\mu_{M_t^k}})^2} \tag{4.2}$$

where $m_{t,d}^k$ are the realizations of the key metric M_t^k at time window t, in day d.

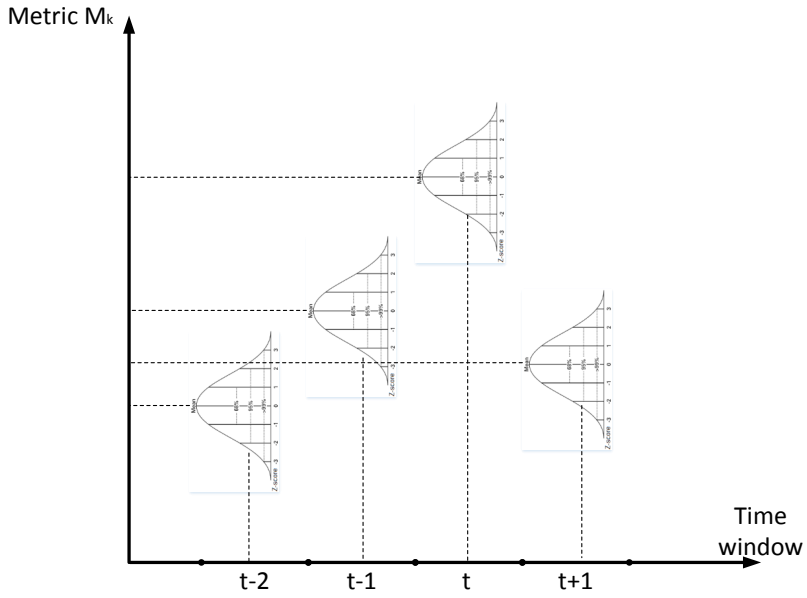


Figure 4-2 – Normal distribution of key metrics M^k at each time window.

In the detection and validation stage, in a specific time window t, the final decision to raise the alarm or not is made by a test hypothesis with the null hypothesis $H_{0,t}$ as the event is normal, the alternative hypothesis $H_{1,t}$ is there is an anomaly in this time window. The test hypothesis is presented in equation 4.3:

$$\begin{cases} H_{0,t}: & \text{all decision } H^k_{0,t} \text{ holds} \\ H_{1,t}: & \text{at least one } H^k_{1,t} \text{ is detected} \end{cases} \tag{4.3}$$

The test in equation 4.3 depends on multiple smaller tests, corresponding to each key metric M^k_t. Each smaller test has the null hypothesis $H^k_{0_t}$ as the new observed value of a key metric $m^k_{t,d}$ is normal, and the alternative hypothesis $H^k_{1_t}$ as abnormal. The test hypothesis is presented in equation 4.4:

$$\begin{cases} H^k_{0,t}: & m^k_{t,d} \text{ is normal} \\ H^k_{1,t}: & m^k_{t,d} \text{ is abnormal} \end{cases} \tag{4.4}$$

The null hypothesis is hold when $m^k_{t,d} \leq TM^k_t$. The type I error (false negative)

happens when an observed key metric value is abnormal but it is smaller than the threshold. Type II error (false positive) happens when the situation is normal but the observed key metric value exceeds the threshold. The threshold TM_t^k is defined by:

$$TM_t^k = \mu_{\hat{M_t^k}} + \alpha * \sigma_{\hat{M_t^k}} \tag{4.5}$$

where α is a tuning parameter. The value α is chosen as in [37], where the probability of a value falling outside the confidence interval TM_t^k is at most $1/\alpha^2$, which is at most 6.25% for $\alpha = 4$. The smaller the value of α is, the higher the number of true positive (good detection) and false negative (fail to detect) are. The combination of multiple classifiers that correspond to multiple key metrics explain the name of this classifier.

One-class SVM classifier

This classifier is proposed as a alternative approach to analyze jointly the key metrics instead of separately. And the data used in this classifier is a high dimensional point combined from the key metric values at the same day and time of multiple key metrics. In particular, each sample of this point is a row in the vector of key metric X_t. So the high dimensional point is the vector $v_{t,d} = [t, m_{t,d}^1, m_{t,d}^2, ..., m_{t,d}^k, ..., m_{t,d}^K]$. With this type of data, many classifiers normally applied for points data could be applied to detect the sample that is different from the major popularity.

After extracting the vectors of key metrics, we need to normalize them because each key metrics have a different scale. Many normalization methods have been discussed in [60]. Two popular methods are min-max and z-score.

In the min-max normalization, the normalized values are given by equation 4.6, where we normalize the vector $v_{t,d}$ at the key metric M_t^k.

$$\tilde{m_{t,d}^k} = \frac{m_{t,d}^k - \min_{\forall d \in D, \forall t \in T} m_{t,d}^k}{\max_{\forall d \in D, \forall t \in T} m_{t,d}^k - \min_{\forall d \in D, \forall t \in T} m_{t,d}^k} \tag{4.6}$$

with D is the total day in the dataset, and T is the number of time window. The z-score normalization, also called the standardization calculates the normalized scores as equation

4.7.

$$\tilde{m^k_{t,d}} = \frac{m^k_{t,d} - \mu_{\hat{M^k_t}}}{\sigma_{\hat{M^k_t}}} \tag{4.7}$$

where $\mu_{\hat{M^k_t}}$ and $\sigma_{\hat{M^k_t}}$ are estimated mean and deviation of M^k_t explained in equation 4.1 and 4.2. In the proposed method approach, we use the z-score normalization because the maximum value of the key metric is impossible to define. After the normalization step, the vector $v_{t,d}$ becomes $\tilde{v_{t,d}} = [\tilde{t}, \tilde{m^1_{t,d}}, \tilde{m^2_{t,d}}, ..., \tilde{m^k_{t,d}}, ..., \tilde{m^K_{t,d}}]$. We examine which key metric is appropriate for the classifier. A qualified key metric can distinct two classed in the labeled dataset.

The classifier implemented in our approach is the one-class SVM classifier. The training input of this classifier is the vectors $\tilde{v_{t,d}}$ in many days and time windows. Since each instance is a point in high dimensional space, we can use normal kernel such as the RBF kernel explained in section 3.2.4 to compute the kernel matrix for the one-class SVM classifier. The l_2-norm dissimilarity used because the data used are numeric vectors $\tilde{v_{t,d}}$. This classifier has two hyperparameters σ of the RBF kernel and ν of the one-class SVM model. In the detection process, new vectors of key metrics are normalized using the same μ and σ of the training process. Then the one-class SVM classifier is used to predict the label of new vectors.

4.3 VIRTUALIS - office environment

4.3.1 Description

The first case that uses the AED method is the working office case at ThereSIS laboratory within the VIRTUALIS project. This method aims to strengthen current security systems to identify attackers. The sensor system has been described in section 2.2. In this section, we discuss the key metrics, the time windows and the critical area selection, and then we present the experimental results using the simulated dataset and the real dataset provided by the industrial partners as described in section 2.2. Figure 4-3 depicts the simulation plan of the Thales building. The simulation plan is showed because it covers a larger zone of the building than the real plan. In fact the real plan of VIRTUALIS contains

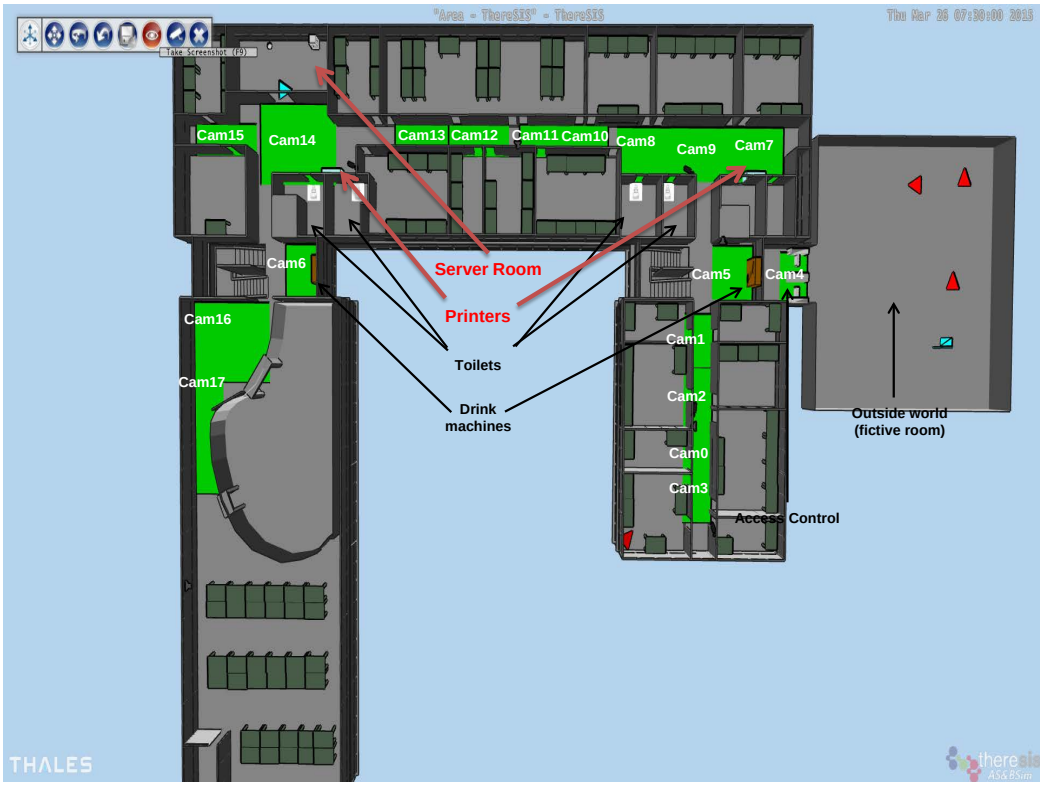


Figure 4-3 – Modelization of the Thales site.

only the zone between the camera 10 and camera 6. It leads to the fact that the printer 2 (right) is not exist in the real dataset.

4.3.2 Zone partition

In the VIRTUALIS scenario, the Thales building can be divided into areas according to the walls between the offices. The technique AED concentrates on the most critical areas in the building, where vulnerable elements are located. Two items were used in our experiment: the printers and the server room whose positions in the building are depicted in Figure 4-3.

4.3.3 Key metrics

In this scenario, the types of key metrics in a controlled zone that were chosen for experiments are:

- *Number of visits*: the aim is to control the number of persons in a selected zone. This metric counts the number of visits in a time window, assigning a visit to the time window of the entering moment.

- *Occupation rate*: the aim is to control the percentage of the time window such that at least one person occupies the area.

- *Average duration of a visit*: this key metric controls the average duration of stay. The event is assigned to the time window of the entering moment.

- *Maximum duration of a visit*: this key metric controls the maximum duration of all the visits in a time window. The events are assigned to the time window of the entering moment.

- *Number of printings*: it controls the number of printing commands in the time window.

4.3.4 Experimental results

The ensemble classifier and the one-class SVM classifier have been trained with data of the normal days in the training set. Then the classifiers were used to detect abnormal events in the day which contains the attacking scenario. The experiments were done with both simulated dataset and real dataset.

Experiments with simulated dataset

The simulated dataset contains 11 days, where nine days are normal and two days contain abnormal events. There are two attack scenarios, the first one is an intruder disguised as an employee, who tried to hack into the information system through the printer. This event happened in the first attacking day. The second attack is an intrusion in the server room which happens on the second attacking day.

The first experiment is consisted in the ensemble classifier using simulated dataset. In this experiment, we applied the method for three areas: the server room and two printer zones. The position of these areas in the building are presented in Figure 4-3. We applied

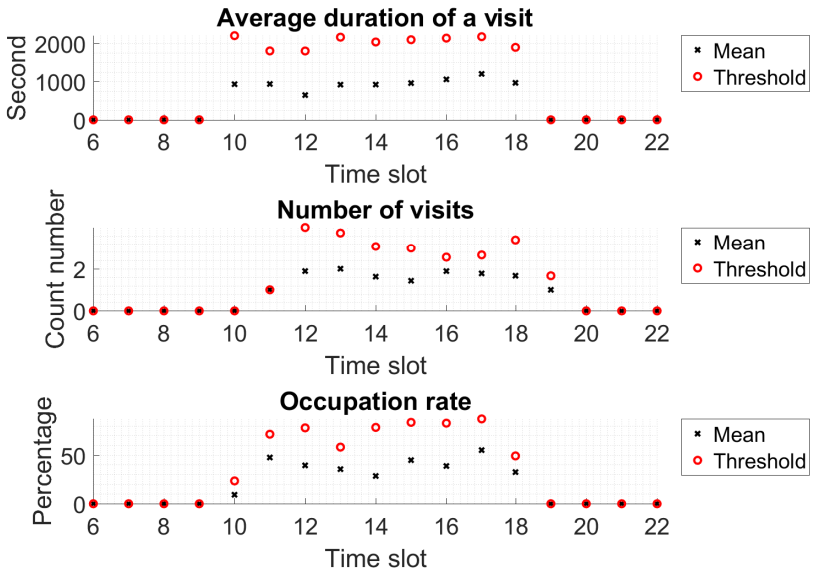


Figure 4-4 – Training results at the Server room.

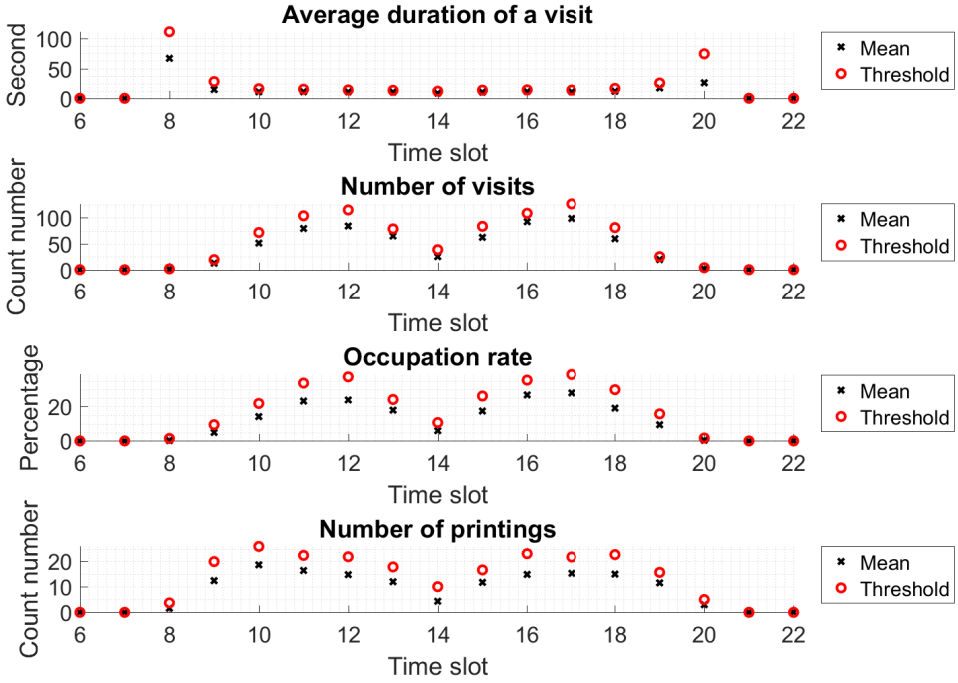


Figure 4-5 – Training results at printer 1.

4 types of key metric: *average duration*, *number of visits*, *occupation rate* and *number of printings*. The key metric type *number of printings* is only usable in printer zones. All the time windows in this experiment have a duration and a shift of 1 hour, so a day is divided into 24 slots.

The Figure 4-4 represents the training result in the server room for the different types of key metrics. The Figure 4-5 shows the training results at the printer 1 (left). This printer is placed on a passage where many employees pass. The Figure 4-6 represents the key metrics at the printer 2 (right), which is placed in a corner of the building.

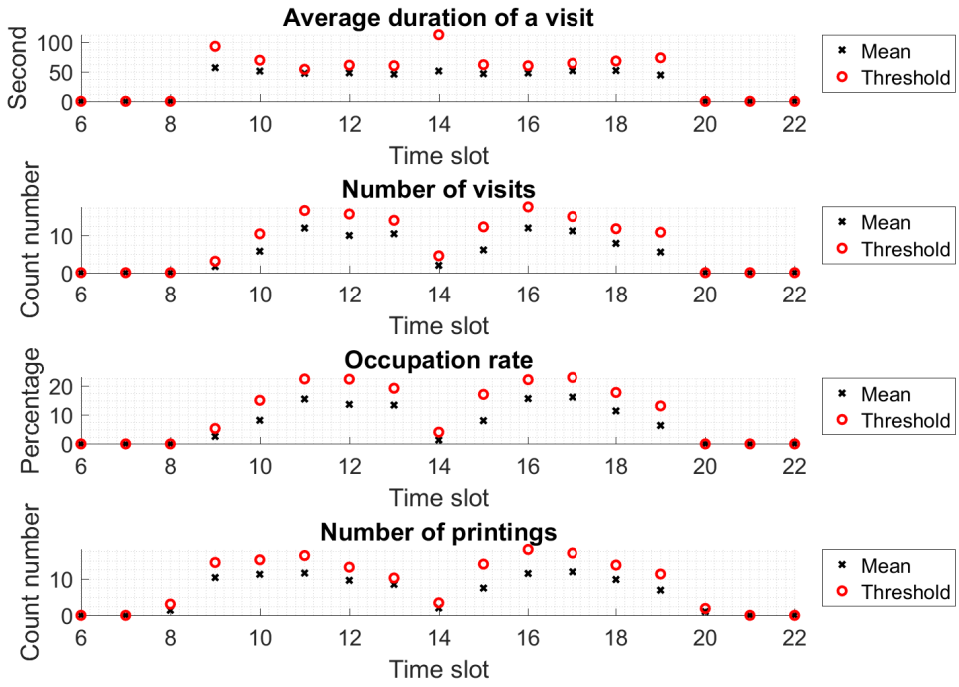


Figure 4-6 – Training results at printer 2.

The training results show different values for these zones. The average duration of a visit in the server room is longer than in a printer zone. This fact is mostly because the server room is a working place, only administrators are authorized to access the server room, and the printer zone is put in a public space where many people can access easily. Next, the number of visits at printer 1 is higher than in server room and printer 2. It's because of the limited access to the server room and the position of the printer 2. The mean

of key metrics in respect with the time window have the shape of a mixture Gaussian due to the working schedule at Thales, where employees start to work at around 8 a.m, have lunch at noon and finish working at approximately 19 p.m.

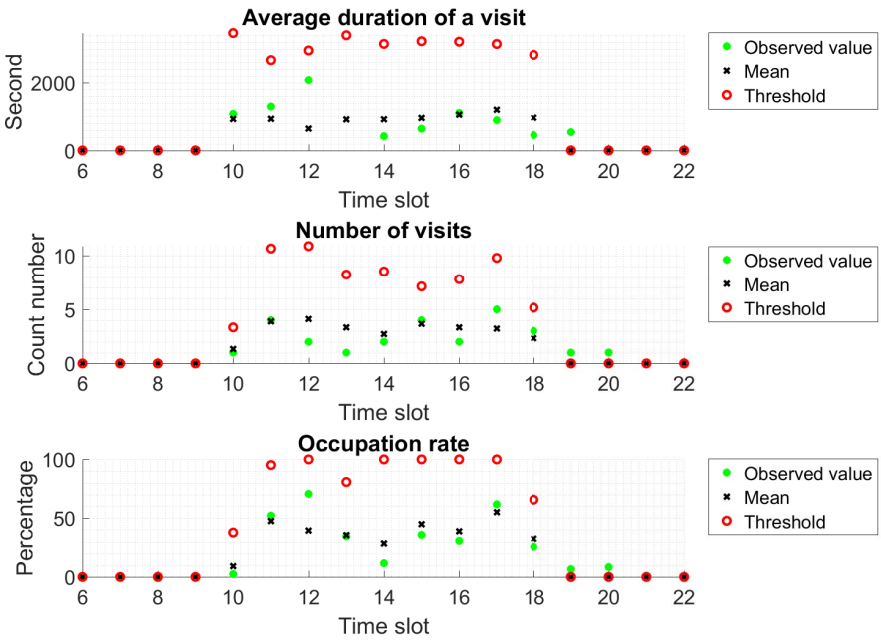


Figure 4-7 – Detection results in server room for the first attacking day.

The model has been tested using the first day with the attacking scenario in the server room. An alert is raised if the key metric of the test day exceeds the corresponding threshold. Figure 4-7 shows the detection result in the server room. The three key metrics *average duration*, *number of visits* and *occupation rate* showed an abnormal access to the server room between 18 and 19 o'clock. The key metrics *number of visits* and *occupation rate* have an abnormal sample at time slot 20 (19 to 20 o'clock) while the key metric *average duration of a visit* does not, because the key metric *average duration of a visit* only counts an event for the time slot when it starts. The other key metrics count an event for all the time slots that event belongs to. The attack event in the server room started at 18:55 and end at 19:05, which leads to this result.

The Figure 4-8 represents the result for the printer 2. It corresponds to the attacking scenario on the second attacking day, where the hacker accesses the information system

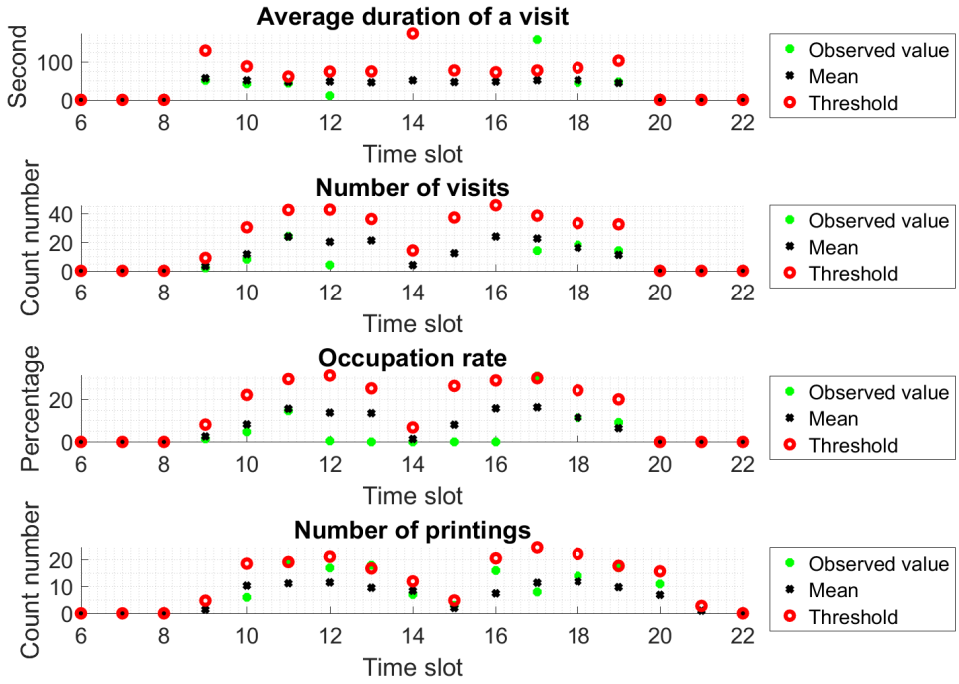


Figure 4-8 – Detection results at printer 2 for the second attacking day.

through the printer. The AED technique detects an abnormal *average duration of a visit* and *occupation rate* in the time window 17 (between 16 and 17 o'clock). Some false alarms are caught in the key metric *number of printing*.

These results demonstrate that two attacks were detected by different key metrics. The number of false alarm could be reduced by increasing the threshold. Thus the tuning of the parameter α need more dataset for training. The training and validation results of this experiment show values of mean and threshold which is quite perfect because the data is simulated by software using the working schedule at Thales. Furthermore nine days of training set provide a dataset with values which are generally representative, except for the training to train the model.

The second experiment is consisted in the determination of a one-class SVM classifier using vectors $v_{t,d}$ in normal training days. The selected types of key metrics were composed of *average duration*, *number of visits*, *occupation rate*, and a feature of time window. A small

shift of 60 seconds and a window duration of 30 minutes are set to obtain many samples for the training set because we only had nine days for training. In the training process, the vectors of key metrics are normalized to $\tilde{v_{t,d}}$ before computing the kernel matrix. With such time window parameters, we obtained 4008 instances using nine days of the dataset. The validation set contains 467 key metrics vectors of the second attack day, and they are normalized with the same parameters in the training process. The vectors of the time windows which contain abnormal trajectories were labeled as abnormal. An one-class SVM model was trained and validated using these datasets. A Receiver Operation Characteristic (ROC) curve is obtained by varying the parameter ρ in the model.

The Figure 4-9 presents the ROC curves for three values of σ. $\nu = 0.1$ is fixed in this experiment. This figure shows an excellent performance of the classifier with $\sigma = 1$. This result is obtained because the attacking scenario in the simulated dataset is an easy case, where most of the key metric in the ensemble classifier can detect correctly the abnormal events with almost no false alarm.

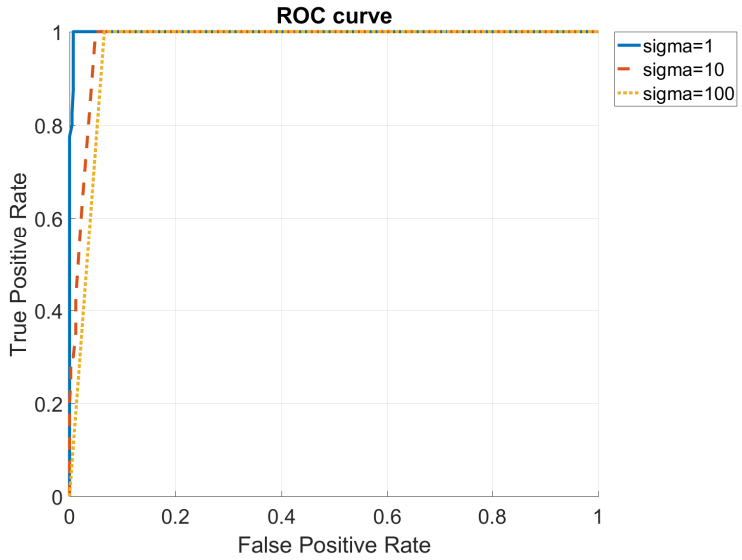


Figure 4-9 – ROC curve of the one-class SVM's performance on the simulated dataset.

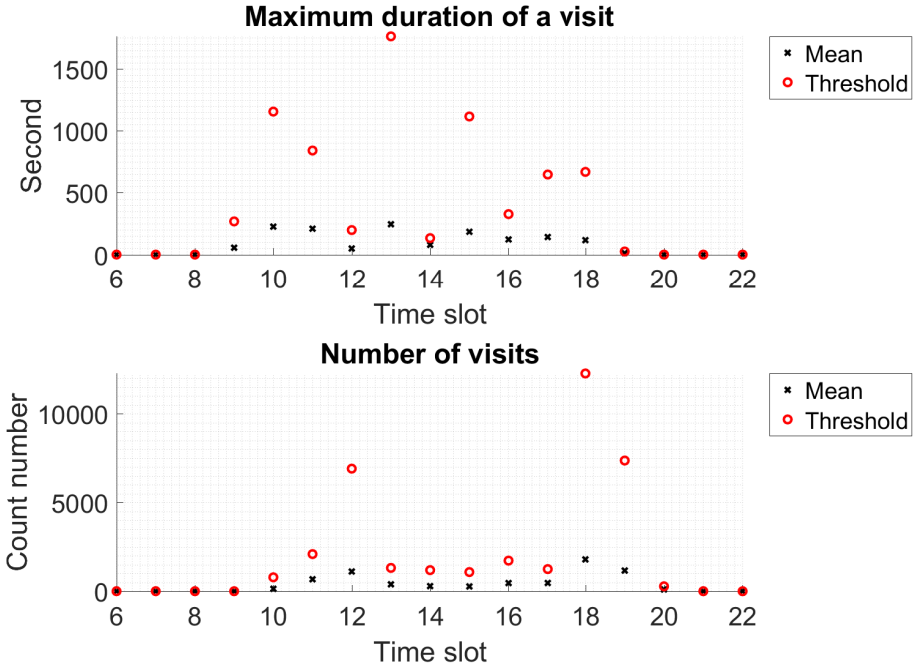


Figure 4-10 – Training result of the ensemble classifier on the real dataset.

Experiments with real dataset

The experiments using the simulated dataset were replicated with the real dataset. There are four days in this dataset, so we use three days which contains normal situation for training and one day for test. The surveillance area in this experiment is the 0.5-meter square zone around the printer 1. Two experiments with this dataset were done, the same as with simulated dataset, where the first experiment establishes the ensemble classifier and the second experiment builds the one-class SVM classifier.

The ensemble classifier was trained with two key metrics *maximum duration of a visit at the printer 1* and *number of visits at the printer 1*. The time window has a duration and a shift equal to one hour. The Figure 4-10 represents the training result of the two key metrics, i.e. it shows the mean value of the random variables and the threshold to be applied on the observed samples to take the decision of the two key metrics.

The pedestrian tracking task of a person, which should be a full trajectory is interrupted into smaller trajectories because of the imperfect algorithm which is a difficult task.

That's why the number of visits is high. The training threshold is not as nice as the one with simulated data.

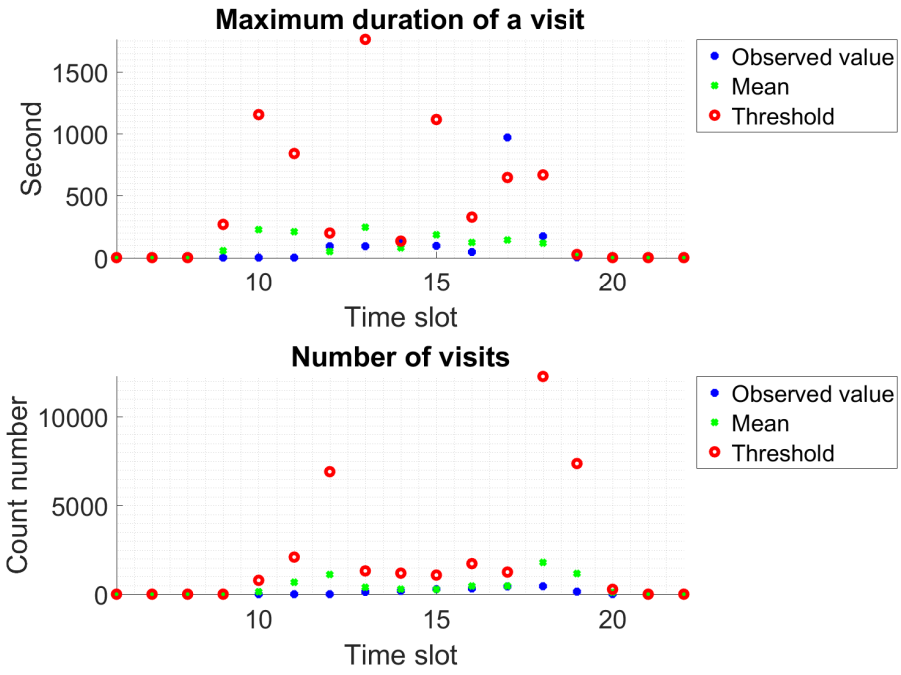


Figure 4-11 – Detection result of the ensemble classifier on the real dataset.

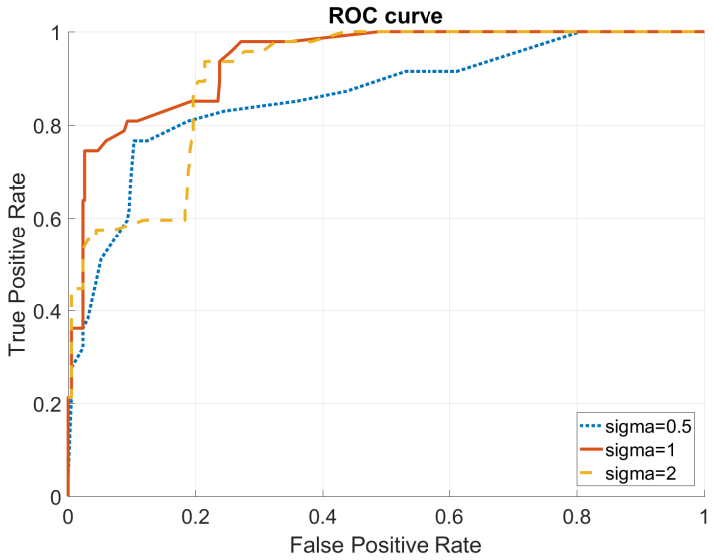


Figure 4-12 – ROC curve of one-class SVM classifier on the real dataset.

The detection result for the day with an attack is given on Figure 4-11. According

to the scenario, an attacker hacks the printer at 16:30 for 10 minutes to steal information. On the figure, it can be seen that an abnormal event is detected in the 17^{th} time slot by the key metric *maximum duration of a visit*.

In the second experiment, a one-class SVM model was trained with 1992 vectors of data extracted from normal days and validated with 464 vectors of key metrics extracted from the unusual day which includes the attack in the server room. The time window has the same parameter as in the experiment with the simulated dataset. Vectors of key metrics compose of 4 features: *maximum duration*, *occupation rate*, *number of visits* and *time window*.

Figure 4-12 presents the ROC curve of the one-class SVM's detection result with three different values for σ, and $\nu = 0.1$ is fixed. The ROC curve shows a less attractive performance than the ROC curve in the simulated dataset, but the result of the model is still considerable with a detection rate at 50% with almost no false alarm (0.5%) with $\sigma = 1$.

This section has presented the implementation details of the working office case in VIRTUALIS project, the AED method performed well on both VIRTUALIS datasets. The following section presents the shopping center case of ATC project.

4.4 ATC - shopping center environment

4.4.1 Description

This section presents the result of the the AED method for the ATC case. Due to the difference in the environment and the dataset between the ATC case and the VIRTUALIS case, some tasks such as zone selection and key metrics definition are modified to suit the ATC scenario. The experiments in this section are similar to VIRTUALIS, i.e. an ensemble classifier and a one-class SVM classifier are trained with unlabeled data and validated on a labeled dataset.

4.4.2 Zone partition

In this case, there is no natural way, in particular no walls, to divide the whole building into areas, and there was no critical element in the building. According to the map of ATC shopping mall, two main regions are the square and the corridor as in Figure 2-7. Because the square connects to a railway station and a dock, people tend to stay in the square to wait for a train or ship, and sometimes exhibition is organized in this area. On the contrary, in the corridor, people tend walking rather than staying because it is a passage that connects many locations. The tracking task in this scenario was good, so the pedestrians are tracked with high accuracy. There is no vulnerable element to concentrate on. The AED technique was applied to the main areas in the ATC case.

4.4.3 Key metrics

The trajectories in ATC dataset contain some additional features such as the height of the person, the angle of the moving direction or the angle between the face and the body as presented in section 2.3. These features allow to extract more types of key metrics, which are:

- *Number of visits*: the number of trajectories.
- *Average duration*: the average duration of a visit.
- *Average height*: the average of the pedestrians' height measurement. This key metric could be used to analyze the categories of pedestrians in the area. For example, a Japanese adult has an average height of 1.6 meters, so when an observed average height of 1.2 meters allows to conclude that there are many children in the building.
- *Average speed*: the average speed of the pedestrians. This key metric allows to observe when the pedestrians are moving faster than usual.
- *Variance of motion angle*: obtained from the feature angle of movement, which describes the angle of the body at each instant. The objective is detecting the pedestrian's inconsistency in movement.

4.4.4 Experimental results

The experiments are divided into three parts. First, we analyzed the key metrics in the dataset, between working days and weekend days, and between the corridor and the square area. Second, we constructed an ensemble classifier that is trained with normal days, and is validated with an abnormal day (day that contains trajectories labeled as abnormal). Third, we construct an one-class SVM classifier using key metric vectors of multiple days for training and use key metric vectors of an abnormal day to validate it.

The labeled dataset contains trajectories from two days 11/11/2012 and 18/11/2012. So to create the validation set for the two classifiers, we used labeled trajectories (both normal and abnormal) from the day 11/11/2012, and merged them with other trajectories of the same day. The vectors of key metrics were computed using this set.

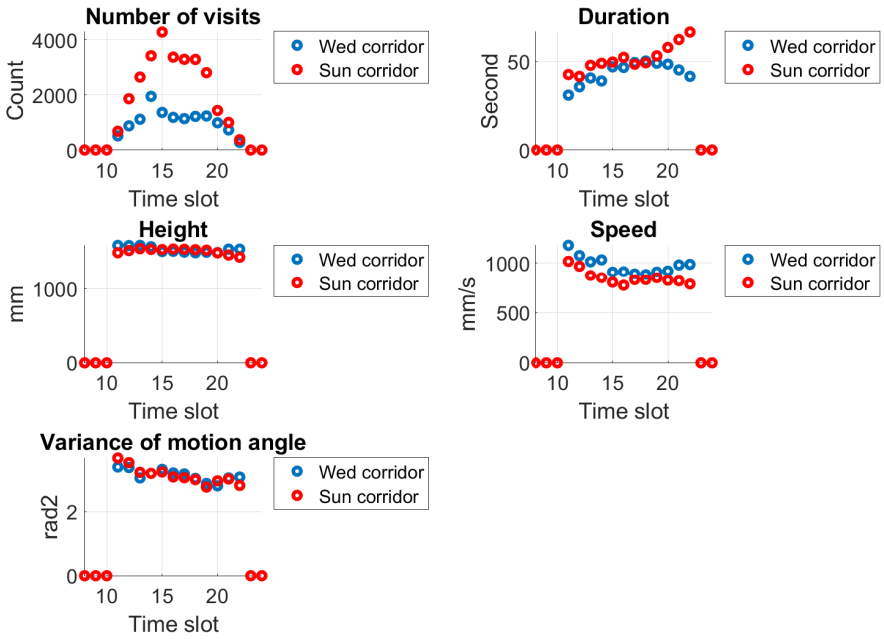


Figure 4-13 – Mean of key metrics on Wednesday (blue) and Sunday(red) at the corridor.

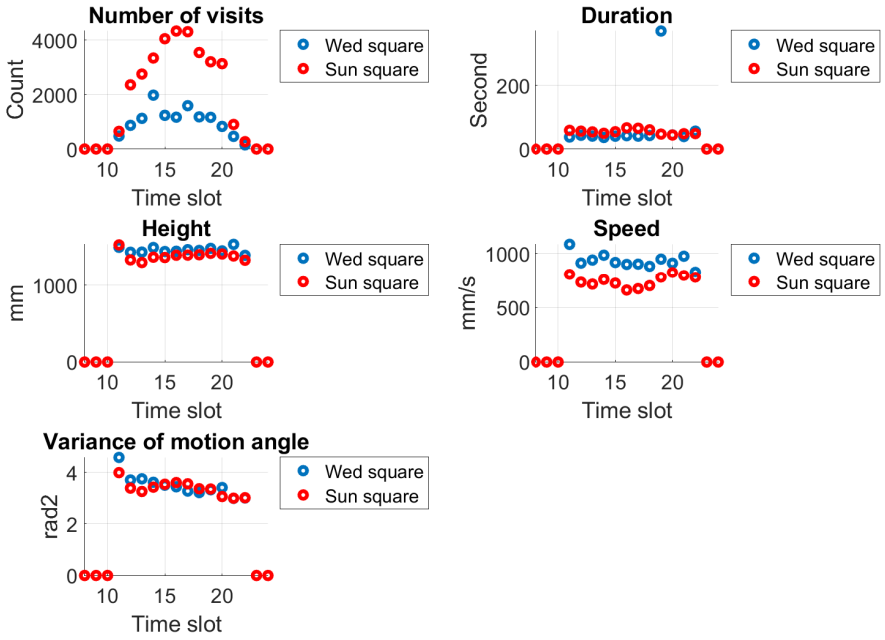


Figure 4-14 – Mean of key metrics on Wednesday (blue) and Sunday(red) at the square.

Key metric analysis

The key metrics analysis was performed in the two regions of the ATC shopping center: the square and the corridor. The key metrics that we used have been described in section 4.4.3. First, we analyzed the difference of key metrics between Wednesday (working day) and Sunday (non-working day).

The figure 4-13 presents the mean of five key metrics in respect with the time along the day in the case of a Wednesday and a Sunday in the corridor. In this figure, we can see a higher number of visits on Sunday than Wednesday. This can be explained by a large number of visitors on the weekend, while during the week the place is mostly occupied by the people working in the building. We can also notice a higher average moving speed on Wednesday than Sunday.

The figure 4-14 compares means of the key metrics between Sunday and Wednesday in the square area. The number of visits on Sundays exceeds the number of visits on Wednesdays, the duration of stay on Wednesday shows a peak at 19:00 due to the ferry

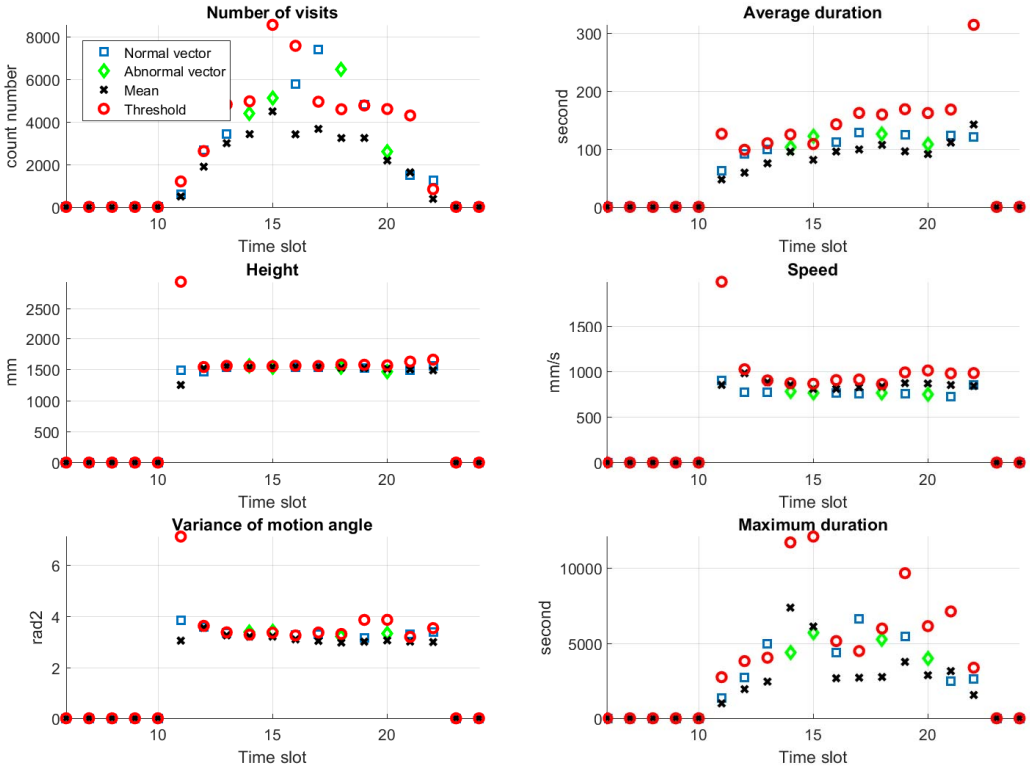


Figure 4-15 – The training and detection result of the ensemble classifier on the ATC dataset.

departure time, which corresponds to the people waiting inside the area before boarding.

Ensemble classifier

The figure 4-15 presents the training and detection result of the ensemble classifiers using vectors of key metrics. At each time slot, the mean and threshold was computed using unlabeled data from 3 Sundays in November and December 2012. The test day is the Sunday 11/11/2012 that contains labeled abnormal trajectories in the dataset. The chosen area is the corridor. As explained before, the vectors of key metrics which contain abnormal trajectories are labeled as abnormal. So the normal and abnormal vectors of key metric in the test day is showed in this figure to represent the detection result. By looking at this figure, one can easily count number of good detection, false alarms, etc.

The key metrics that are applied in this experiment are: *number of visits*, *average duration*, *height*, *speed*, *variance of motion angle*, *maximum duration*. The key metric *maximum duration* was added to replace the key metric *variance of face angle* because the feature *face angle* is not available in the labeled trajectories. The time window in this experiment is kept simple by setting a duration and a shift to one hour.

One-class SVM classifier

The second experiment consisted of building a one-class SVM classifier using 6 Sundays for training and the Sunday 11/11/2012 which contains labeled trajectories was chosen for validation. Vectors of key metrics were extracted from these datasets by computing key metrics in 30 minutes time windows with a shift of 1 minute. The selected key metrics for this experiment are: *average duration*, *average speed*, *maximum speed*, *maximum motion angle*. These key metrics are chosen because they can separate normal and abnormal vector of key metric. We have 4112 vectors of key metrics in the training set and 676 vectors of key metrics in the validation set. The objective was to detect abnormal key metric vectors which contain the abnormal trajectories. The evaluation metric is the ROC curve as for VIRTUALIS project.

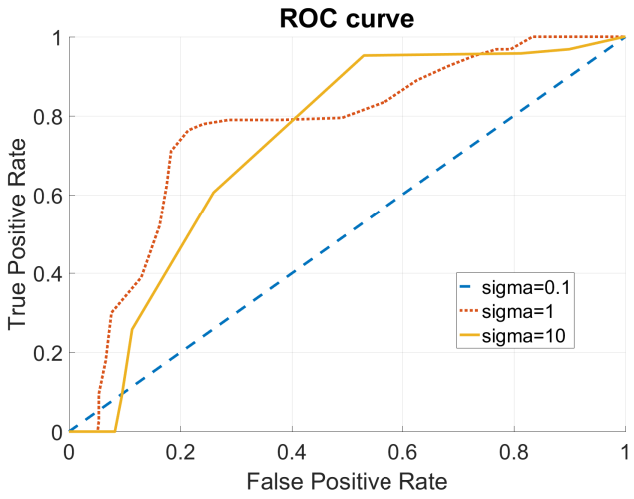


Figure 4-16 – ROC curve of the one-class SVM classifier using vector of key metrics.

The figure 4-16 presents the ROC curves of the model using three values of σ and a fixed $\nu = 0.01$. The reason for the poor detection result of this model is that there

are many abnormal events alike in the training set. Moreover, when the vector of key metric is computed, the normal and abnormal trajectories are merged. For example, we mixed the labeled trajectories on 11/11/2012 with other unlabeled trajectories of the same day to create the validation set. So the effect of the labeled abnormal trajectories to the vector of key metric is reduced because many other trajectories were put in together. This phenomenon is explained with Figure 4-15, where each feature of abnormal key metric vectors did not exceed the threshold. So when they are combined to key metric vectors, the classifier faced a lot of difficulties to classify them.

4.5 Conclusion

In this chapter, the technique AED was presented to detect abnormal events in a zone inside a building. This technique analyzes multiple sub-trajectories in a predefined area in the building. Vectors of key metrics in a time window were introduced and extracted from these sub-trajectories. Two types of classifiers were built using the vectors of key metrics to train the model: an ensemble classifier which detect individual abnormal key metrics in a time window and a one-class SVM classifier which detects jointly abnormal vector of key metrics.

In VIRTUALIS scenario, both simulated dataset and real dataset were implemented for the detection test, and both classifiers gave good detection result with a low false alarm rate. This result proved the ability of this method to detect abnormal state at an area in the building, considering the available datasets.

The ATC dataset is not adapted for the study of security application. An expert created the labeled set of this dataset with the meaning of a person who gets lost. In the unlabeled training set, there is a considerable amount of trajectories similar to the labeled abnormal trajectories. For these reasons, the performance of the classifiers in this dataset has some limitations.

In the method ATD which is presented in chapter 5, this issue affects the performance of the method on ATC dataset compared to the VIRTUALIS dataset. However, its performance is slightly better because the ATD method classifies individual trajectories,

not the combination of multiple trajectories like the AED method.

Chapter 5

Abnormal trajectories detection using One-class SVM model

5.1 Introduction

In this chapter, we discuss the *Abnormal Trajectory Detection (ATD)* method. The motivation of this method came from the idea of creating an unsupervised anomaly detection model trained from the normal trajectories and using this model to detect the trajectories that are very different from the training set. However, with the limitation of the trajectory which contains points of movement as discussed in section 3.3.3, the detection task for this type of data could be challenging. Therefore, instead of using directly the trajectory, we extract a new kind of sequence called sequence of activity from these trajectories. This type of sequence is used in the proposed model.

As in Chapter 4, our method will be applied to VIRTUALIS and ATC cases. In this chapter, the common methodology is presented first. Then the implementation for each case is described. For each application, the sequence extraction process and the corresponding dissimilarity function are presented. Three kernels GDTW, DTA, GA which are specialized for one-class SVM model and sequence data are applied to compare their performances. The experiments show that our model can detect abnormal sequences with a low false alarm rate.

5.2 Methodology

With this methodology, the detector must be trained and can be used to infer detection of new data. First, we define a training dataset to train the one-class SVM detector. The dataset contains sequences extracted from camera observations. There are three main processes in this method: training, validation, and detection process. The training process is offline. After the detector is trained, it is validated on a labeled dataset. This process aims to choose the optimal hyperparameters for the detector. The trained detector is deployed to the security system to detect abnormal sequences. The detection process is performed in online mode.

In the training process, we extract sequences of activities from raw trajectories of pedestrians in typical days. The first step is to split the map of the building into multiple sub-areas. After that, activity at each sub-area is induced using the trajectories movement points at that sub-area. And finally, sequence of activities are formed by connecting the activities extracted from the same trajectory. So the whole trajectory is mapped into a sequence of activity through this sequence extraction process. Then we cut these sequences into smaller parts based on a time window with a maximum duration. After the sequence extraction step, we obtain a dataset which contains several sequences of activities with a limited duration. Then we train the classifier with this dataset. We applied a one-class SVM classifier. As our dataset contains sequences, we applied the three types of kernel GDTW, DTA, GA specified for sequence that were introduced in chapter 3.2.4. These kernels need a dissimilarity measure to compare two activities, so a custom sequence dissimilarity measure is proposed. Combining the dissimilarity measure and the kernel methods, we obtain a Gram matrix of the data to feed in the one-class SVM model.

In the validation stage, a set of labeled trajectories is used to test the trained detector. The sequence extraction is done for this set. Then we compute the Gram matrix between the sequences in the training set and the set of validation sequences. Finally, the detector predicts the labels of the validation set using the Gram matrix to identify the abnormal sequence. The evaluation metric is based on the accuracy, the good detection rate and the false alarm rate.

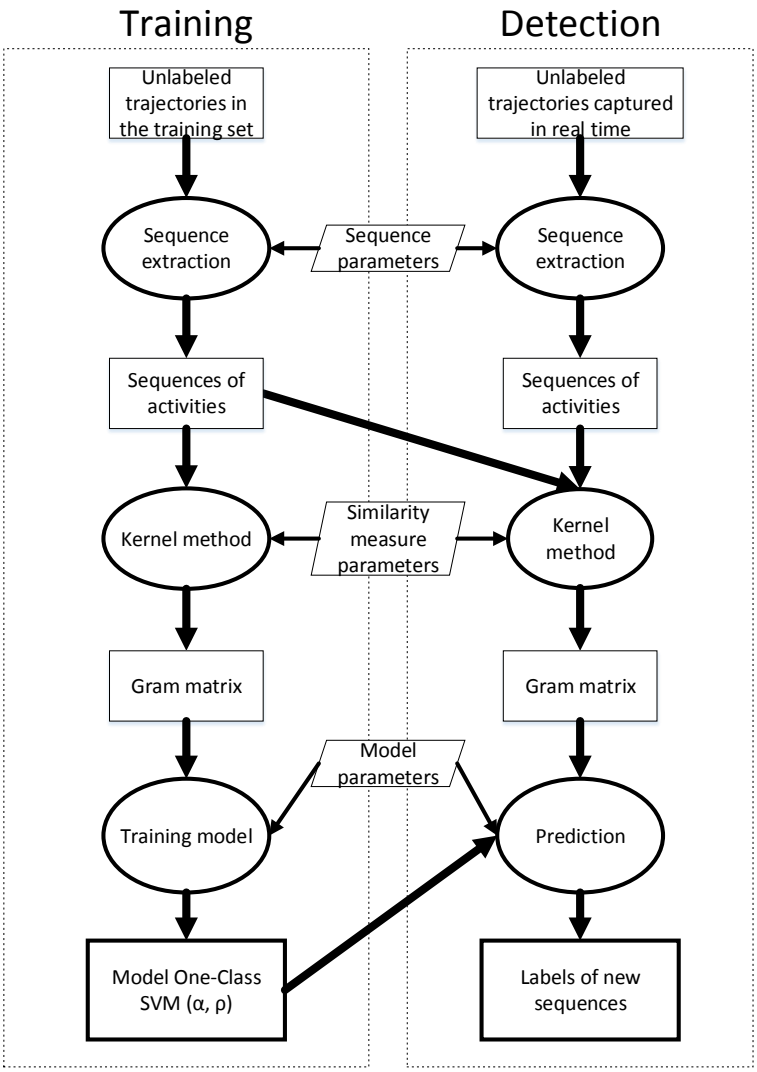


Figure 5-1 – Methodology of the ATD technique.

The detection process is similar to the validation process, the only difference is that input sequences are extracted from real-time trajectories. The detection could be repeated periodically after each period of 5 minutes for example. The training and detection process is presented in Figure 5-1. The validation process is similar to the detection, but one needs to replace the input with labeled trajectories. Three types of hyperparameters have to be determined: sequence parameters, dissimilarity measure parameters, and model parameters. The sequence parameters are fixed for all of the processes, and the two others are calibrated to have the optimal classification result on the validation set. These hyperparameters are changed depending on the scenario, and they will be presented with more detail in

each case's implementation. In this manuscript, the experiments include the results of the training and evaluation process. The detection stage was implemented at the demonstration at ThereSIS laboratory at Thales site in the VIRTUALIS project, and there is no detection stage for the ATC case.

5.3 VIRTUALIS

5.3.1 Description

In this section, we apply the proposed approach to detect abnormal trajectories for the working office building scenario. This section describes more detail about the sequence construction process and the custom dissimilarity function.

The experiments are based on simulated data because the real data faced an issue with people re-identification as mentioned in chapter 2. Typically, when a person enters an office and then gets out, the system recognizes that these trajectories have two different subjects. Therefore the whole trajectory of a person since he enters the building until his exit is cut into multiple fractions. So we can not extract sequences of activities from trajectories with the real dataset.

There are several days of typical activity in the simulated dataset and two days which contains three trajectories of the attackers. So we create our model on an unlabeled dataset and then test the model on a labeled data composed of normal and abnormal trajectories (attacking scenarios).

5.3.2 Sequence construction

This section provides an overview of the sequence creation process. In the office scenario, the building is naturally divided into separating sub-areas by the walls. The cameras supervise some sub-areas such as the corridors or the area around the printer, and there are sub-areas out of camera's reach such as inside the office of employees. The graph representation of Thales simulated building is presented in Figure 5-2. Each node in this

graph represents a specific zone, where people can do activities. The edges link the zones together, which show the possibilities to move from one node to its neighbors. The green nodes in Figure 5-2 display the monitored sub-areas, and the white nodes are areas without cameras. The red points are critical elements such as server or printers.

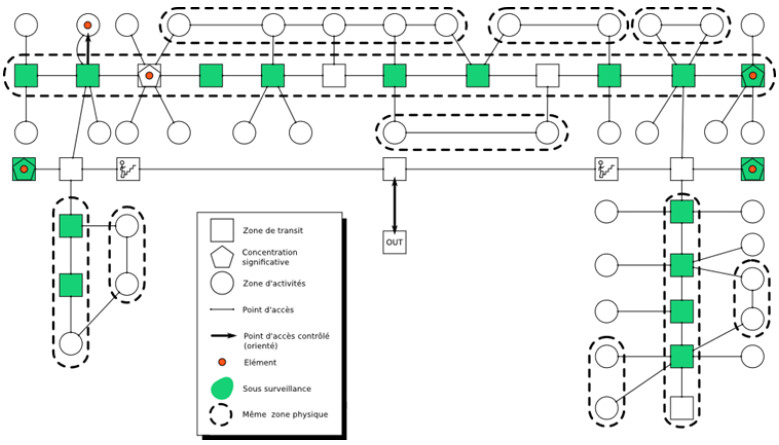


Figure 5-2 – Graph representation of Thales building

The next part of this section will introduce the definition of an activity and the activity extraction process which transform initial trajectories to sequences of activities.

Activity

The definition of an activity used in this approach needs to be specified.

Definition 5.3.1. *Activity is an action of a person in a specific place. An activity is created when a person stays at the same place for longer than a predefined threshold. This place can be both under camera's surveillance or not. The type of the sub-area infers the type of activity. Each activity is characterized by a vector* $A \in R$, $A = [A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}]$ *where the four features are the ID of the person, the type of location, the beginning time and the activity duration respectively.*

Note that the type of activity could be inferred precisely if additional information is available such as room planning for a meeting or working schedule of each individual.

In this application, three types of activity are considered. The first type is the activity in a monitored place, where cameras record the sub-trajectory in that area. An action is created when the total duration of the sub-trajectory exceed a predefined time called $T_{sup-act}$. This type of activity is called sup-act (supervised activity). The second type of activity is the activity in non-monitored sub-areas. This kind of activity is created using a gap in a trajectory if the duration of that gap is larger than another threshold called $T_{unsup-act}$. This type of activity is called unsup-act (unsupervised activity). For example, a trajectory of an employee continues from the entrance to the door of an office, and then 1 hour later this trajectory is continued from the same door, so the activity of this employee in that office is created (supposed that $T_{unsup-act} < 1$ hour). The last type is called E2-activity (Entrance and Exit activity). This type is created when a person enters or leaves an area according to card reader's information. The *type of location* feature is the place where the

Table 5.1 – Example of an activity

$A^{(1)}$	$A^{(2)}$	$A^{(3)}$	$A^{(4)}$
ID	Location's type	Time	Duration (s)
101	Printer	9:00:00	300
102	Office	15:20:00	3600
103	Passage	08:00:00	20

action happens, and it also indicates the activity in that place. Thus, the position of the location is ignored (office 100 and office 101 is considered the same), only the functionality of the site is used to describe the activity in that place. For example, the typical activity in an office is working, and the ordinary activity in the printing zone is printing, so instead of specifying the action by saying someone is working in the office or printing some document, we assume that someone is doing something in the office or the printing room. These location's types are used to indicate these actions. *Time* is the beginning instant of the activity, and *duration* specifies how long the activity lasts. Three activities of three people are described in Table 5.1. Each *ID* is unique for each employee in the building.

Sequence of activities

The series of activities of each person are combined into sequences of activities. Since we only have offline sequences (completed sequences which already happened), but

our purpose is designing a model for online detection, we used a sliding time window to cut the offline sequences into overlapped time ordered sub-sequences in order to create online sequences from offline sequences. A sliding time window has two hyperparameters: *maximum duration* L and the *shift* Sh. These hyperparameters are defined to limit the total duration of each sequence so that the information is kept but not too complex and not too intrusive for individual privacy issue - typically 2 or 3 hours maximum. The sequence of activities is represented as S. $S \in R^{N\times 4} : S = [A_1, A_2, ..., A_i, ..., A_N]$. A_i means the i^{th} activity of sequence S where N is the number of activities in sequence $S, i \in [1, N]$. Note that two sequences may have the same duration but a different number of activities. An example of a sequence S with $N = 3$ and $L = 2h7m$ is presented in Table 5.2.

Table 5.2 – Example of a sequence of activities

S	ID	Location's type	Time	Duration (s)
A_1	101	Coffee room	9:00:00	360
A_2	101	Office	9:06:00	7200
A_3	101	Printer	11:07:00	60

This table represents a sequence composed of three activities. The first activity is in the coffee room for approximately 5 minutes, then this person works in his office for 2 hours, and then he prints some document at the printer. The time between the end of an activity and the beginning of the next activity is after a few seconds, used for moving between the sub-areas.

The proposed detector is based on one-class SVM model to classify abnormal sequences. This classifier creates a hyperplane in a high dimensional space to separate anomalies from normal instances. So a dissimilarity measure to compare two instances is critical for this method. The sequence structure presented above is a complex data type where each sequence contains multiple activities, and features of an activity is a mixture of numeric and non-numeric values. In this situation, we can not apply a formal solution like combining a standard dissimilarity measure such as Euclidean distance with RBF kernel to our problem. This characteristic of data forces us to propose a custom dissimilarity measure combined with kernel method to compute the kernel matrix between sequences. This dissimilarity measure is used to compare two activities, which is the fundamental element to calculate the kernel matrix. Moreover, three different kernels will be applied to our approach.

5.3.3 Custom dissimilarity measure and kernel method

In this section, a method that calculated the kernel matrix between sequences of activities is proposed. This method includes a function to compare two activities and a sequence alignment algorithm to calculate the distance between two sequences with different numbers of activities. First, a custom dissimilarity function is defined to calculate the distance between two activities. This function is composed of four parts: ID cost, location's type cost, time cost and duration cost, in relation to the four characteristics of an activity. The total cost is then found by a linear combination of all these costs. The custom dissimilarity function $DissAct$ is presented:

$$DissAct(A_i, A_j) = \sum_{k=1}^{M} \theta_k C_k(A_i^{(k)}, A_j^{(k)}) \tag{5.1}$$

where M is the number of feature in an activity, in VIRTUALIS case, $M = 4$. Each sub-cost function is calculated as follow:

- $C_1(A_i^{(1)}, A_j^{(1)})$ is the distance between two ID, base on their role (engineer, administrator, etc.). In this experiment, we use a simple binary cost based on the comparison of the employee's ID. Cost $C_1(A_i^{(1)}, A_j^{(1)})$ is defined as:

$$\begin{cases} C_1(A_i^{(1)}, A_j^{(1)}) = 0 & \text{if} \quad A_i^{(1)} = A_j^{(1)} \\ C_1(A_i^{(1)}, A_j^{(1)}) = 1 & \text{if} \quad A_i^{(1)} \neq A_j^{(1)} \end{cases} \tag{5.2}$$

- $C_2(A_i^{(2)}, A_j^{(2)})$ is the distance between 2 location's type based on the types of action that can be done in these places. Cost $C_2(A_i^{(2)}, A_j^{(2)})$ is defined as follows:

$$\begin{cases} C_2(A_i^{(2)}, A_j^{(2)}) = 0 & \text{if} \quad A_i^{(2)} = A_j^{(2)} \\ C_2(A_i^{(2)}, A_j^{(2)}) = B & \text{if} \quad A_i^{(2)} \neq A_j^{(2)} \end{cases} \tag{5.3}$$

  where B could have different values defined based on the criticality of the location $A_i^{(2)}$ and $A_j^{(2)}$ (I.e., B is high if $A_i^{(2)}$ or $A_j^{(2)}$ is a critical place such as server room).

- $C_3(A_i^{(3)}, A_j^{(3)})$ is the difference between the beginning time of two events. This cost

is proportional to time shift between A_i and A_j when the difference is larger than a threshold $\gamma^{(3)}$. For smaller difference, the cost is set to 0. Otherwise, it increases as a linear function with the slope $s = 1$. This function is similar to the shift recifier (relu) function in Figure 5-3.

$$C_3(A_i^{(3)}, A_j^{(3)}) = \max((|A_i^{(3)} - A_j^{(3)}|) - \gamma^{(3)}, 0) \tag{5.4}$$

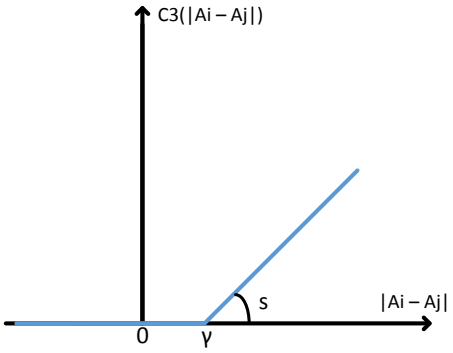


Figure 5-3 – Distance function.

- $C_4(A_i^{(4)}, A_j^{(4)})$ is the difference between the duration of two events. Cost $C_4(A_i^{(4)}, A_j^{(4)})$ has the same analytical form as cost $C_3(A_i^{(3)}, A_j^{(3)})$, with threshold $\gamma^{(4)}$. The function C_3 and C_4 is illustrated in Figure 5-3.

From our point of view, the cost that indicates the difference between two location's type has more impact on the dissimilarity function than others factors because the type of damage that can be caused in each place is different. For example, something abnormal in the server room is more critical than in the restroom.

The hyperparameter θ in equation 5.1 plays an important role which directly affects the model's performance. By definition, θ defines the weight of each feature to the total cost between two activities. For example, if we want two activities that happen at a very different time of the day to have high distance, we can set θ_3 large in comparison to others θ. The same rule is applied to other features. However, we do not want a specific feature dominant the rest.

In fact, the detail costs C_k between each feature's type have their range of value. So vector θ is chosen so that the term $\theta_k C_k$ of all features vary in the same range value (i.e. [0,1]). To achieve this constraint, a random set G of activities is selected from a set of random sequences. The calculation of θ_k is the inverse estimated average cost as presented in 5.5.

$$\theta_k = G(G-1)(\sum_{i,j} C_k(A_i^{(k)}, A_j^{(k)}))^{-1} \tag{5.5}$$

with $k \in [1, M]$, where $\frac{(G(G-1))}{2}$ is number of couple of activities computed in this context.

After defining the dissimilarity function to compare two activities, one needs to calculate kernel used to feed in the one-class SVM classifier. To explain this step, we recall the optimization problem of the one-class SVM classifier for sequences using kernel matrix:

$$\begin{aligned} \min_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i \alpha_j \mathcal{K}(S_i, S_j) \\ s.t. \quad & 0 \leq \alpha_i \leq \frac{1}{\nu m}, i \in [1, m] \\ & \sum_{i=1}^{m} \alpha_i = 1 \end{aligned} \tag{5.6}$$

The discriminant function is expressed as:

$$f(S_{new}) = \sum_{i=1}^{m} \alpha_i \mathcal{K}(S_i, S_{new}) + \rho \tag{5.7}$$

where S_i is the i^{th} sequence in the training set and S_{new} is the new sequence for prevision. In equation 5.6, $\mathcal{K}(S_i, S_j)$ is the kernel value for sequence S_i and S_j of the chosen kernel which replace the mapping function ϕ as discussed in Section 3.2.4. Three kernels GDTW, DTA and GA kernel presented in section 3.2.4 are compared.

In the proposed approach, three types of hyperparameter are needed: sequence parameters, dissimilarity parameters and model parameters. Sequence parameters are used

for the sequence extraction process such as L, Sh, $T_{sup-act}$, $T_{unsup-act}$. This type of parameter defines the activity. Similarity parameters of the dissimilarity measure: B, $\gamma^{(3)}$, $\gamma^{(4)}$ and kernel parameters σ. These parameters define the way we measure the distance between two events. These parameters include the parameter θ which controls the weight of each activity's feature on the dissimilarity function. Model parameters of the classifiers (ν of the one-class SVM model or C of SVM model).

5.3.4 Experimental results

In this section, our proposed anomaly detection model performs the classification task on the simulated dataset of the VIRTUALIS project. There are 11 days in this dataset, nine days normal that have no attack and two days which contain the anomalous event. So we use a sample set of the trajectories of the first nine days to train our model. A set of 300 sequences is randomly selected from the dataset of first nine days, and the Gram matrices of three kernels GDTW, DTA and GA are calculated based on this set of sequences.

To evaluate the performance of our model, we prepared a validation set of 300 normal sequences and 119 abnormal sequences. The normal sequences were selected randomly from sequences of normal users in the two days that have abnormal events. The abnormal sequences came from three attacking events. As explained in section 5.3.2, the offline sequences are cut into overlapped sub-sequences to create online sequences. So we could have 119 sequences labeled as abnormal with only three attacking events (offline sequences).

After preparing the validation set, we calculate the Gram matrix between the validation set and the training set, and then we give labels for sequences in the validation set. The ROC curve is used to evaluate the performance of our model on different types of kernel. For each couple of hyperparameter σ and ν, a ROC is obtained by varying the value of ρ in the one-class SVM model. This action will change the distance from the detection boundary to the origin in the high dimensional feature space. By looking at the ROC, we can evaluate the trade-off between the good detection availability and the number of false detection of the model.

In the following part of this section, we present the detection results of the proposed

unsupervised classifiers of three kernels on the validation set. The evaluation metric that we used is the accuracy, the true positive rate and the false positive rate. The last part is the sequence analysis which is performed to have a deeper understanding of the detection result.

One-class SVM detection

The ROC curve presents the true positive rate on the vertical axis and the false alarm rate on the horizontal axis. A good ROC curve approaches the upper left corner, which constitutes a high detection rate with a low false positive rate. The ROC curve is unique for a couple of hyperparameter σ and ν, and is obtained by varying the parameter ρ in the one-class SVM model.

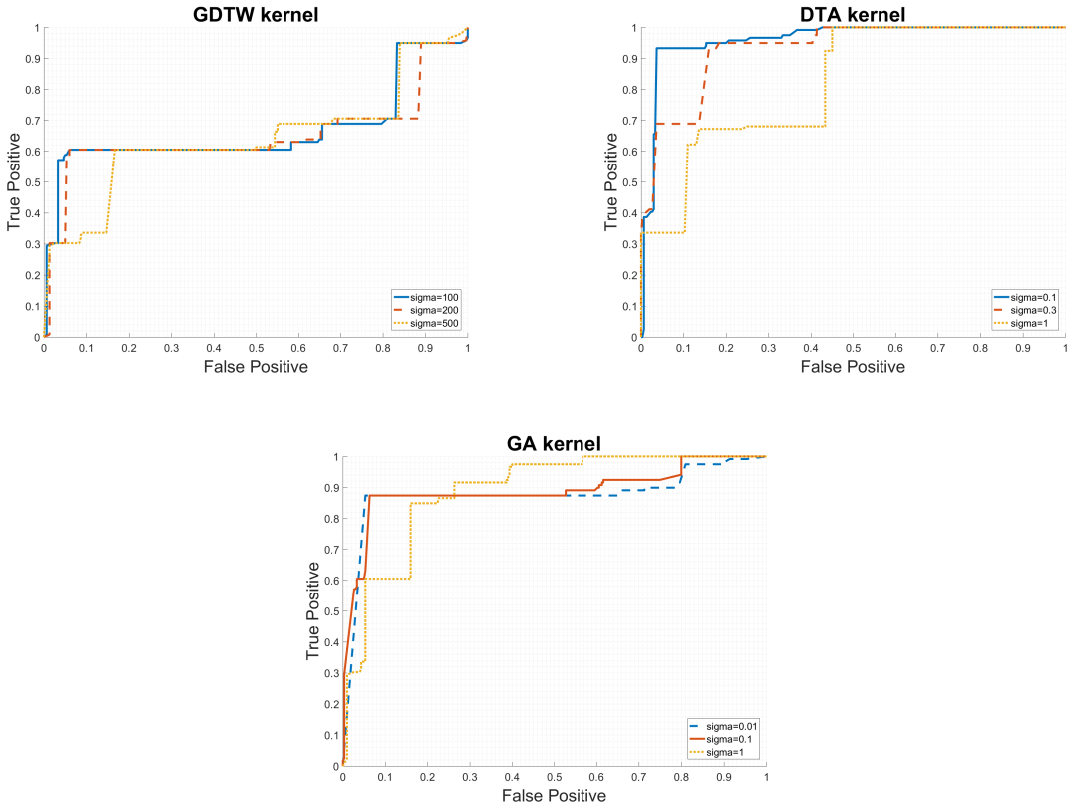


Figure 5-4 – ROC curves of multiple kernels of the one-class SVM model. From left to right, and then down are the ROC curves of three kernels GDTW, DTA, GA respectively.

In Figure 5-4, multiple ROC curves is presented for each kernel corresponding to

different value of σ. $\nu = 0.1$ is fixed. Due to these ROC curves, the GA kernel seems to perform best, and the second winner is the DTA kernel, the last is GDTW kernel. Table 5.3 presents the best performance of each kernel.

Table 5.3 – Performance of different kernels of the One-class SVM classifier.

Kernel	σ	ν	Acc	TPR	FPR
GDTW	200	0.1	0.84	0.6050	0.06
DTAK	0.1	0.1	0.8807	0.6555	**0.03**
GA	0.01	0.1	**0.926**	**0.8739**	0.0533

Table 5.3 presents the highest accuracy value of each kernel with a fixed $\nu = 0.1$. The optimal σ changes for each kernel. This table shares the same opinion with the ROC curve, where GA kernel has the best accuracy, and the false positive is kept very low (5%). There are three abnormal events, two of them could be detected easier than the event which corresponds to a preparation step of the attacker. It means a detection rate (TPR) of 0.6 is acceptable for this experiment where 2 out of 3 attacks are detected, and all of the kernels seems to satisfy this.

Sequences analysis

This section analyzes the sequences detected as abnormal by our detector, including good detection and false alarm. According to the attacking scenario, there are three attacking events. First, an example of two good detection sequences are presented in table 5.4 and table 5.5.

Table 5.4 – Attacking sequences at the printer.

ID type	Time (s)	Duration (s)	Zone
Visitor	57627	0	In/Out
Visitor	57633	867	Printer room
Visitor	58508	0	In/Out

The first sequence in table 5.4 happened in the first abnormal day. It is a sequence of a visitor entered the building, then he accessed the printer zone for 15 minutes and then leaves. This sequence is classified as abnormal because of long duration in the printer area of a visitor.

Table 5.5 – Attacking sequence in the server room.

ID type	Time (s)	Duration (s)	Zone
Visitor	68127	0	In/Out
Visitor	68129	23	Passage
Visitor	68161	540	Server room
Visitor	68710	24	Passage
Visitor	68735	0	In/Out

In the second sequence which happened in the second abnormal day in table 5.5, a visitor enters the building at about 7PM (68400s), then he sneaked into the Server room directly, he stayed there for nearly 10 minutes and then left. This sequence is probably classified as abnormal due to the presence duration in the server room late in the day of a visitor.

Table 5.6 – False negative.

ID type	Time (s)	Duration (s)	Zone
Visitor	35210	0	In/Out
Visitor	35213	15	Passage
Visitor	35230	3540	Office
Visitor	38770	17	Passage
Visitor	38790	0	In/Out

Table 5.7 – False alarm.

ID type	Time (s)	Duration (s)	Zone
Cleaning woman	27867	0	In/Out
Cleaning woman	27871	100	Office 1
Cleaning woman	27976	370	Office 2
Cleaning woman	28350	180	Office 3
Cleaning woman	28534	98	Office 4
Cleaning woman	28635	219	Office 5
Cleaning woman	28875	156	Office 6
Cleaning woman	29036	186	Office 7
Cleaning woman	29236	399	Restroom 1
Cleaning woman	29639	225	Restroom 2

The third sequence in table 5.6 is a false negative (false detection). This sequence happened in the first abnormal day, before the sequence in table 5.4. The purpose of this attacker was plugging a malware USB into the printer. However, his presence in an area was not long enough to form an activity in the printer zone. This sequence was classified as normal because all activities in this sequence are normal, and the hacker was very careful

not to leave any atypical trace.

The last sequence in table 5.7 is an example of false positive which is a sequence of a cleaning woman. This sequence is detected as abnormal because sequences of cleaning woman is a minority in our training set. So the classifier may have left it as outliers.

In the following section, we present the implementation of this technique for the ATC dataset which corresponds to the shopping center scenario. In this scenario, the environment is different from the office environment in the VIRTUALIS scenario. So some modifications are introduced to adapt this method.

5.4 ATC

5.4.1 Description

The second part of this chapter presents the implementation of the ATD technique for the shopping mall scenario in the ATC project. In this project, the re-identification problem is better solved using a system of 3D-camera, and most of the place in the building is under camera's surveillance. So the trajectory of each person is fully tracked inside the building. However, we can not bring the whole method ATD defined in the office scenario above to solve this problem. The first reason is there are no natural division sub-areas in this area, so the graph segmentation process has to be modified. Second, the dataset of this scenario provides more features in the raw trajectory, so the definition of activity and the dissimilarity function was changed due to the different structure of trajectory in the ATC dataset.

A set of normal and abnormal trajectories selected by an expert in [18] is used to evaluate the model in this scenario. Two experiments were done with this dataset. The first experiment compares our model and author's model performance, so a supervised SVM model is trained based on the labeled dataset. The hyperparameter of this model is defined using Kernel Alignment method presented in section 3.2.4. The second experiment is same as the VIRTUALIS's, where an unsupervised one-class SVM model is trained using unlabeled sequences and tested on the labeled sequences. This approach concentrates on

the analysis of the ATC dataset, and the performance to classify labeled trajectories.

5.4.2 Sequence construction

Area division

The graph segmentation process creates a graph out of the building's plan. First, a grid of points is formed inside the boundary of the building. Then the whole building is divided into a desired number of zones (50 zones in our case) with an equal surface. Next, the k-Means algorithm clusters all these points into 50 clusters, and the result is obtained as in Figure 5-5.

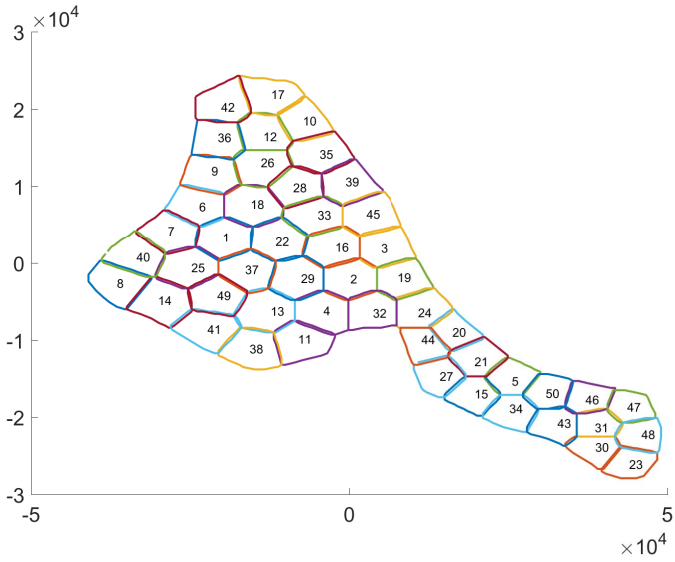


Figure 5-5 – Zone division using K-means clustering.

Figure 5-5 is the result of dividing the whole building in Figure 2-7 into 50 equal size sub-areas, which is a graph with 50 nodes and the edges are the boundary between nodes.

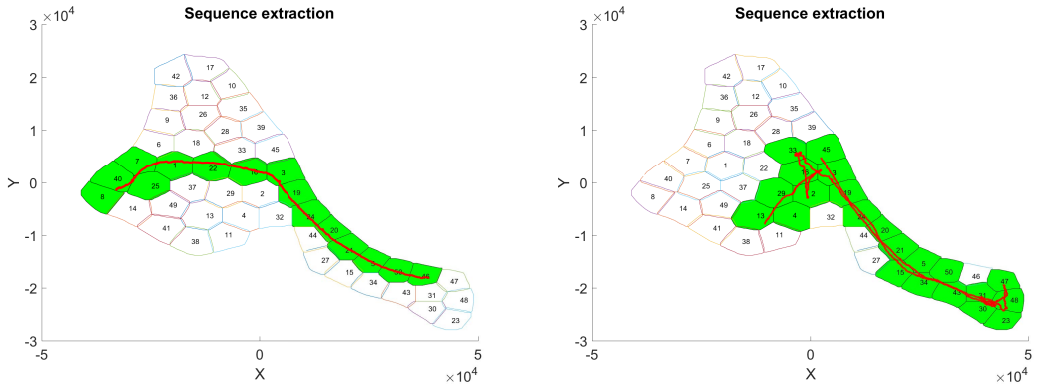
Sequence of activities

After separating zones, we extract sequences of activities from raw trajectories. There is only one type of activity in this scenario compare to 3 types for VIRTUALIS scenario

because the whole area is divided into equivalent nodes. The information about the elements in the ATC building is not available so every zones are considered as the same. An activity in a node is created by collecting all points of sub-trajectories in that node. A trajectory crosses a sub-area two times would create two activities. In the ATC case, each activity of this sequence is a vector of 7 features: $A = [A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}, A^{(5)}, A^{(6)}, A^{(7)}]$. The first 4 features is similar to the activity in VIRTUALIS project, where $A^{(1)}$ is the identity of the person, $A^{(2)}$ is the identity of the zone, every zones in ATC case has the same meaning, and the only difference is the location of the zone. $A^{(3)}$ is the entrance time in the zone, $A^{(4)}$ is the duration inside the zone.

Table 5.8 – Example of a sequence of activities in ATC dataset.

SX	$A^{(1)}$	$A^{(2)}$	$A^{(3)}$	$A^{(4)}$	$A^{(5)}$	$A^{(6)}$	$A^{(7)}$
	ID	Location	time	Duration (s)	Speed (mm/s)	Angle (rad)	Ratio
AX_1	101	21	9:00:00	10	860	0.1	0.6
AX_2	101	5	9:00:10	30	1200	2	0.7
AX_3	101	50	9:00:40	25	1000	0.5	0.8



(a) First sequence

(b) Second sequence

Figure 5-6 – Trajectory (red) and sequence of activity (sequence of green sub-areas).

The additional features in the ATC dataset are: $A^{(5)}$ is the average speed in that zone, $A^{(6)}$ indicates the variation of his angle movement, $A^{(7)}$ is the ratio of the distance between the entrance and exit position and his actual moved distance in a zone. To explain these additional features, first we have to recall the raw trajectory's format of the ATC case in Table 2.3. Since a current speed is measured at each observation in a raw trajectory, the average speed feature is calculated by taking the mean of the speed feature ($A^{(5)}$) of observations within a sub-area. The feature angle movement ($A^{(6)}$) is calculated by the

variance of the movement angle feature of a trajectory in a sub-area. The purpose of this feature is to explore the difference between a person who is inconsistent in where he is going and a person who goes straight. The ratio movement ($A^{(7)}$) feature is calculated by taking the percentage of the distance one has actually moved and the length of his passage.

An example of a sequence of activities is presented in table 5.8. The result of the sequence extraction is presented in Figure 5-6, where the trajectories (red line) are extracted to sequences of activities (green connected zones). With this definition of activities, a new dissimilarity function is proposed in the following section.

5.4.3 Custom dissimilarity measure and kernel method

Due to the modification of the sequence definition, the dissimilarity function is also modified to suit the new sequence's structure. Suppose that we have two activities A_1 and A_2 defined as above. The custom dissimilarity function $DissAct$ that compares these two activities are defined similarly to equation 5.1. However, with an addition in an activity's feature such as the angle of movement, the ratio of movement and the average speed, the function is also modified and becomes:

$$DissAct(A_i, A_j) = \sum_{k=1}^{M} \theta_k C_k(A_i^{(k)}, A_j^{(k)}) \tag{5.8}$$

Where M is the number of feature in an activity. In this case $M = 7$. The computation of each cost is as follow:

- $C_1(A_i^{(1)}, A_j^{(1)})$ is the distance between two ID, it defines the distance of two trajectories when their subject is different. Cost $C_1(A_i^{(1)}, A_j^{(1)})$ is defined by:

$$\begin{cases} C_1(A_i^{(1)}, A_j^{(1)}) = 0 & \text{if} \quad A_i^{(1)} = A_j^{(1)} \\ C_1(A_i^{(1)}, A_j^{(1)}) = 1 & \text{if} \quad A_i^{(1)} \neq A_j^{(1)} \end{cases} \tag{5.9}$$

  This scenario is a public place, so the true identity of the pedestrians is unknown. Therefore we only compare if the given identity of two sequences is similar or not.

- $C_2(A_i^{(2)}, A_j^{(2)})$ is the distance between two locations of these two activities, which is the Euclidean distance between two zone center's coordinate.

$$C_2(A_i^{(2)}, A_j^{(2)}) = \sqrt{(X_{A_i^{(2)}} - X_{A_j^{(2)}})^2 + (Y_{A_i^{(2)}} - Y_{A_j^{(2)}})^2} \tag{5.10}$$

  where $X_{A_i^{(2)}}$ and $Y_{A_i^{(2)}}$ are coordinates of the sub-area's center where activity $A_i^{(2)}$ happens, the same for $X_{A_j^{(2)}}$ and $Y_{A_j^{(2)}}$.

- $C_3(A_i^{(3)}, A_j^{(3)})$ is the difference between the beginning time of two events, defined in equation 5.11. Cost $C_3(A_i^{(3)}, A_j^{(3)})$ is proportional to time shift between A_i and A_j when the difference is larger than a threshold $\gamma^{(3)}$. For smaller difference, the cost is set to 0. Otherwise, it increases as a linear function.

$$C_3(A_i^{(3)}, A_j^{(3)}) = \max((|A_i^{(3)} - A_j^{(3)}|) - \gamma^{(3)}, 0) \tag{5.11}$$

- $C_4(A_i^{(4)}, A_j^{(4)})$ is the distance between the duration of two events. Cost $C_4(A_i^{(4)}, A_j^{(4)})$ has the same analytical form as cost $C_3(A_i^{(3)}, A_j^{(3)})$, with threshold $\gamma^{(4)}$.

- $C_5(A_i^{(5)}, A_j^{(5)})$ is the difference between the average movement speed in two zones. $C_6(A_i^{(6)}, A_j^{(6)})$ is the difference between the variation of movement's angle of two events. And $C_7(A_i^{(7)}, A_j^{(7)})$ is the distance between two trajectories's movement ratios. The Manhattan distance in equation 5.12 is used for these cost.

$$C_k(A_i^{(k)}, A_j^{(k)}) = |A_i^{(k)} - A_j^{(k)}| \quad , k = 5, 6, 7 \tag{5.12}$$

After defining the dissimilarity function, three kernels GDTW, DTA, GA are applied to compute the kernel matrix for one-class SVM model as the VIRTUALIS scenario. With the labeled trajectories available in the dataset, the Kernel Alignment algorithm is used to estimate the optimal hyperparameter θ of the dissimilarity function. More explanation of the use of the Kernel Alignment will be explained in the following chapter.

5.4.4 Experimental results

The experiments in this section could be separated into two different parts. In the first experiment, we train a supervised classifier on the same labeled dataset with the author's model in [18]. Then we compare the performance of this classifier with the author's proposed model. This experiment intends to measure the performance of the ATD method by using the same dataset as the author for anomaly detection task.

The second experiment replicates the experiment in VIRTUALIS scenario on the ATC dataset which trains an unsupervised classifier on the unlabeled dataset and test the model on the labeled dataset to compare the performance of different kernels. This experiment studies the capacity of the proposed method for shopping center case.

The experiments in this section are done using two sets of trajectory: labeled and unlabeled dataset. The first one is a labeled trajectory set which contains 29 abnormal and 29 normal trajectories labeled by an expert. The abnormality of a trajectory comes from a person who gets lost and may need help. The second dataset is a set of unlabeled trajectories. This dataset is a subset of the enormous ATC dataset, it contains 300 random trajectories.

Kernel Alignment

Before going to the two principal experiments, we present the result of the KA algorithm to estimate the optimal hyperparameter θ. Different from the VIRTUALIS project where θ was determined to balance the effect of each feature on the cost function. The θ in ATC case is determined by the similarity distance of Kernel Alignment algorithm (equation 3.34).

More specifically, the grid search algorithm is applied to loop through different values of θ to decide if any θ is optimal for the classifier, and the similarity distance of the Kernel Alignment algorithm helps to make the decision. It is done by computing the distance between the Gram matrix of a particular θ computed using a set of labeled sequences and the ideal Gram matrix as discussed in section 3.2.4.

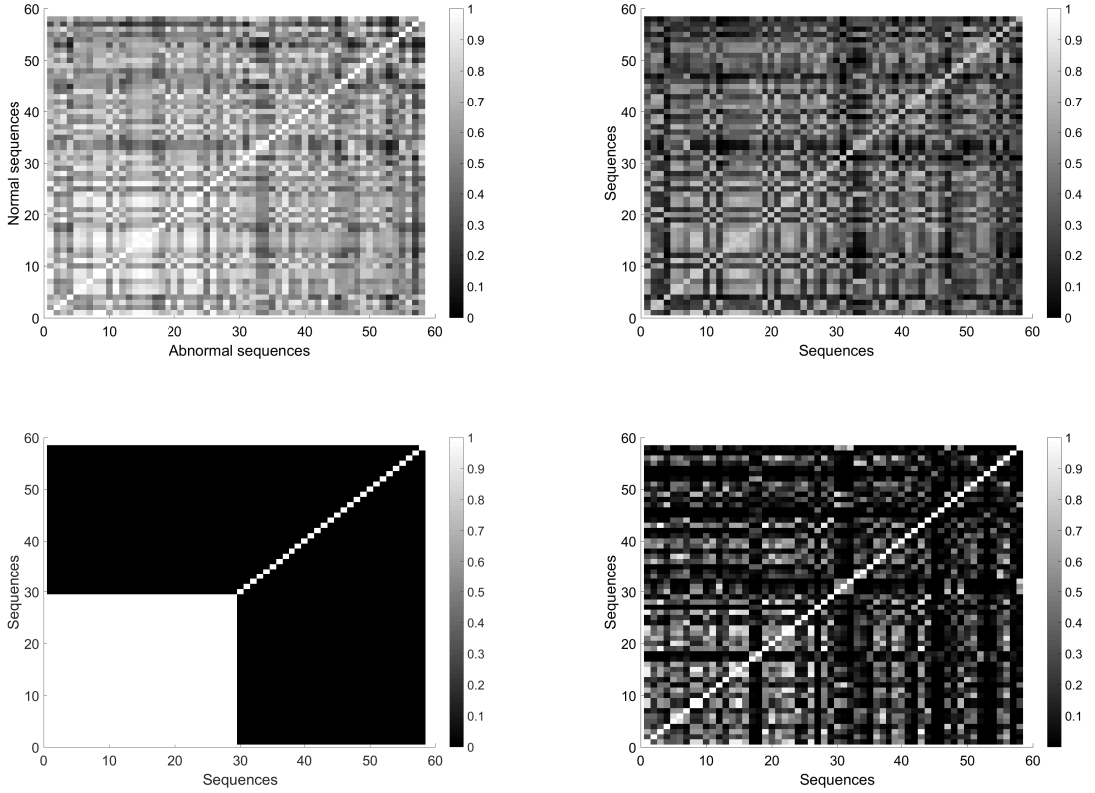


Figure 5-7 – Gram matrix using the parameters determined by the Kernel Alignment algorithm. Starting from top left in clockwise order are three Gram matrix of GDTW kernel, DTA kernel , GA kernel, and lastly is the ideal Gram matrix.

Three types of kernel are applied to compare their best distance. The optimal θ^* corresponds to the minimum similarity distance (equation 3.34) of each kernel is kept. The Gram matrix of three kernels at the optimal θ^* are presented in Figure 5-7 using the labeled dataset. In this figure, the GDTW kernel seems to give the closest format to the desired matrix G. This result is coherent with the kernel alignment distance in Table 5.9.

Table 5.9 – Optimal cost of kernel alignment for different kernels.

Kernel	$d(K, G)$
K_{GDTW}	**1.2011**
K_{DTA}	1.214
K_{GA}	1.2207

In this table, the minimal distance obtained for each kernel is presented. As expected, the GDTW kernel have the minimum distance between three kernels. This result is somehow opposite to the result of the VIRTUALIS scenario where GA kernel performed best. This

result will be verified through the detection experiments.

Leave One Out test for SVM classifier

The first part of this experiment is to use the computed kernel above to build a SVM classifier. The objective is creating a supervised classifier to classify the sequence of activity defined above using the labeled dataset which contains 29 normal sequences and 29 abnormal sequences. This result can also be used to compare with the work in [18]. However, the two experiments are not the same because the author has created a classifier and test on the same dataset. We do not agree with this approach because the model could over-fit with the training dataset.

Due to the small number of labeled sequences, we use the Leave One Out (LOO) technique to avoid the over-fitting phenomenon of the model. The θ^* computed using the kernel alignment method is used for this experiment and the next experiment with the one-class SVM as well. With this θ^* and each couple of hyperparameters σ and C, we apply the LOO technique for the set of 58 sequences, and obtain multiple detection results by varying the hyperparameters σ and C in the SVM model.

Table 5.10 – Performance of different kernels of the SVM classifier.

Kernel	σ	C	Accuracy	TPR	FPR
GDTW	300	100	0.9138	1	0.1724
GDTW	300	10000	0.9138	1	0.1724
DTA	10	10	0.8103	0.7586	0.1379
DTA	1	100	0.8103	0.8276	0.1724
GA	10	0.1	0.8448	0.8276	0.1379
GA	10	100	0.8448	0.8276	0.1379
Brvsvcic [18]			0.9138	0.931	0.103

Table 5.10 presents the performance of the kernels at different couples of hyperparameters σ and C that give the optimal accuracy of each kernel. In this table, the GDTW kernel gives the best accuracy, and comparable to the result of the author in [18] where accuracy is about 90%. Moreover, we have used the LOO test, so our model does not over-fit the dataset. The other kernels (GA and DTA) have very considerable results, and the trade-off between the true detection rate (True Positive Rate (TPR)) and the false alarm

rate (False Positive Rate (FPR)) is showed for different value of σ and C.

One-class SVM test

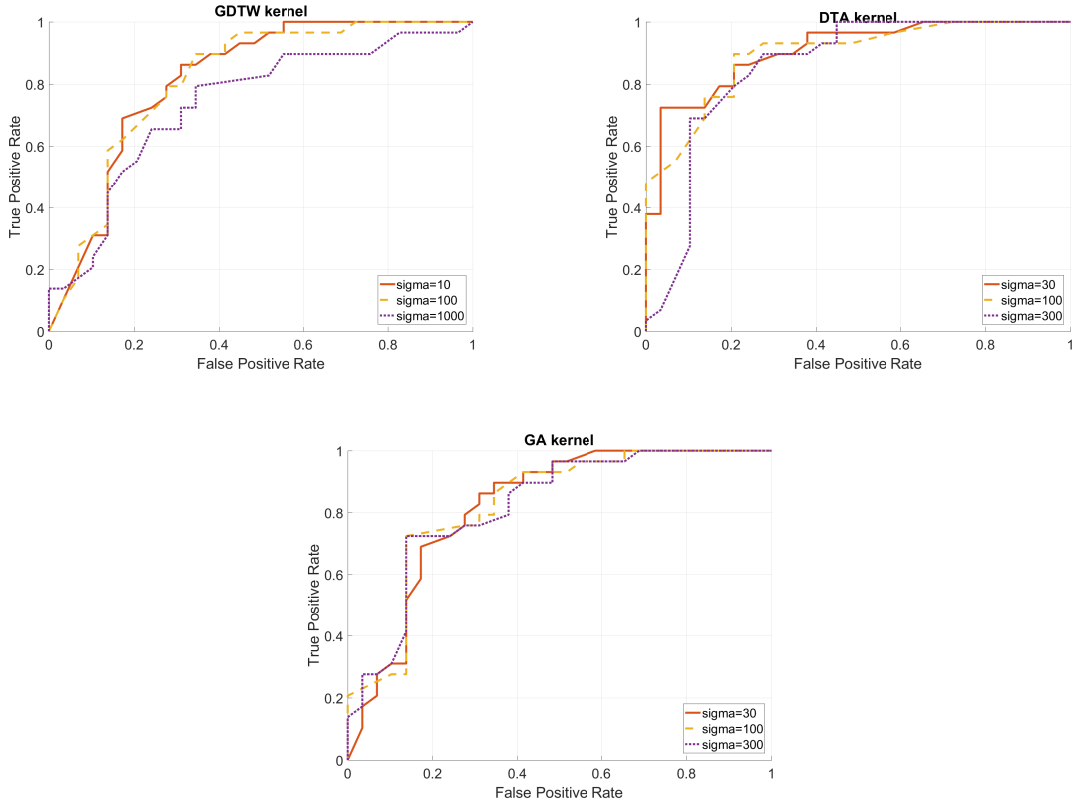


Figure 5-8 – ROC curves of multiple kernels of the one-class SVM model. From left to right, and then down are the ROC curves of three kernels GDTW, DTA, GA respectively.

This experiment trains an unsupervised classifier by unlabeled dataset, and validate the model by the labeled dataset. First, we extract 300 unlabeled sequences from trajectories. Second, we calculate the kernels used to train the classifier. The optimal hyperparameter θ^* of the experiment above was reused. Then we apply a grid search algorithm for σ and ν, and we evaluate the model on the labeled dataset. Figure 5-8 plots the ROC curves with $\nu = 0.1$ and different σ for three chosen kernels. The ROC curve in Figure 5-8 shows that the DTA kernel seems to perform best since it has the highest detection rate ($>$0.4) with very low false alarm.

Error analysis

This section analyzes the result of the classifier on the ATC dataset. The ROC curve in this experiment is not smooth because we only have 58 labeled sequences in total, 29 sequences for each class.

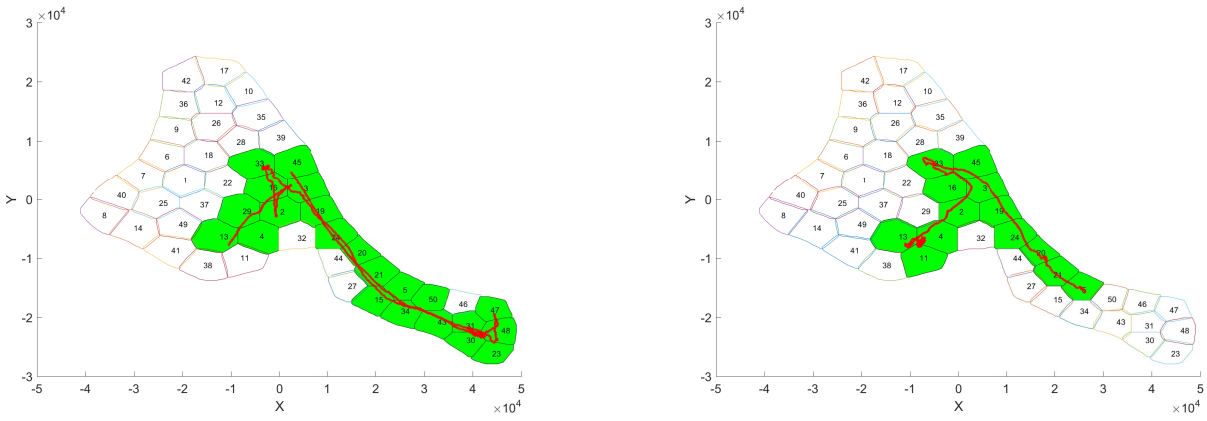


Figure 5-9 – Good detection

Figure 5-9 presents the first abnormal trajectories which were detected when we increase ρ. The abnormality of these sequences is obvious because the indecisive intention of the pedestrian. He walked around inside the building many times.

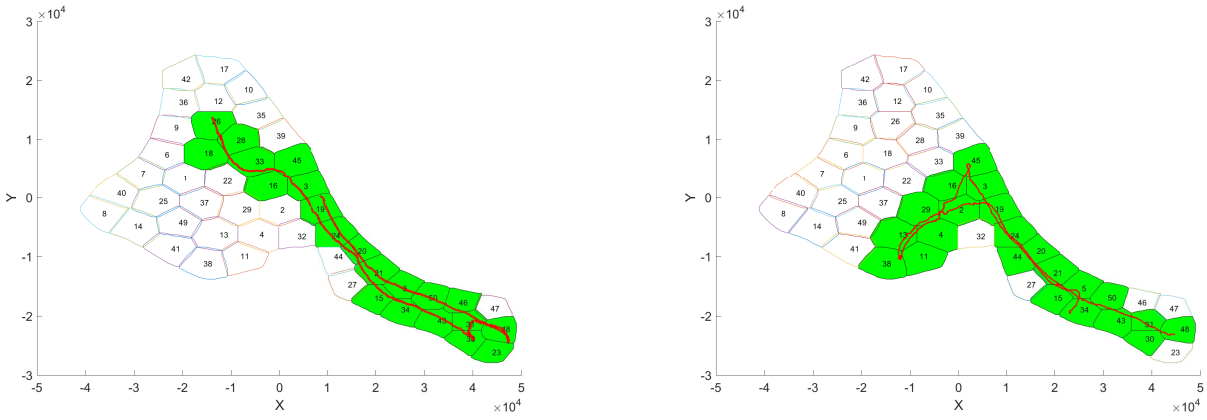


Figure 5-10 – False alarm GA

The false alarms of the classifier are caused by normal sequences that look abnormal. Two of these sequences are sketched in Figure 5-10, they are classified as abnormal due to

the pedestrians' movement which is not consistent. They go around and then return to the position in the first place. However, these sequences are labeled as normal in the dataset.

5.5 Conclusion

In this chapter, we introduced an approach called ATD which processes trajectories as sequences of activities. The objective of this approach is to create a unsupervised classifier to detect abnormal sequence of activities. The classifier is based on one-class SVM model. A method combined by a custom dissimilarity function and three kernel methods for the sequence is applied to compute the distance between sequences and therefore prepare input for the model. This approach is applied to two scenarios, the office in the VIRTUALIS dataset and the shopping center in the ATC dataset.

In the VIRTUALIS scenario, the objective of the detection model is detecting the intrusion in the building. However, we encountered a problem when extracting sequences of activities in the real dataset due to the insufficient of the re-identification task. So for this scenario, the experiments are done using a simulated dataset. Three kernels have been trained for the One-class SVM model. The model was trained on a subset of normal sequences of activities and validated on a labeled set composed of normal and abnormal sequences. Through the experiments, the GA kernel seems to perform best by obtaining the highest accuracy (92.6%), but all kernels can detect at least 60% of the abnormal sequences with a small false alarm rate (0.5%).

In the ATC scenario, abnormal situation to detect corresponds to the person who gets lost in the building. The building has a huge open space which is different from the working office case. For this scenario, we have full trajectories of pedestrians since the re-identification is better. However, the definition of activities is different, and a new dissimilarity function has to be defined. The kernel alignment algorithm is applied to estimate optimal hyperparameter θ of the dissimilarity function. This hyperparameter is then used to create a supervised detection model and an unsupervised model. The supervised detection model is trained to challenge the classification model in [18]. So our model was trained and test on the same dataset of author's model. However, since we do not

agree with the author by training and test the model on the same dataset, we perform the LOO test in our experiment, and the chosen classification model is SVM. The result of this model are comparable to author's paper as presented in table 5.10. The unsupervised model is created to replicate the experiment of VIRTUALIS case, and the objective is analyzing the performance of different kernels. This experiment constructed the one-class SVM model on unlabeled sequences and test on the labeled dataset, and the GDTW kernel performs best in this experiment.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

Security for critical infrastructure has become an important issue for every country in the world. The French government has invested massively in upgrading the security system for critical infrastructure by installing many cameras in public places. However, the bottleneck in these systems are the human operators. So the VIRTUALIS project was initiated to solve this problem by creating an automatic surveillance system. This thesis is created by the collaboration between the M2S team of ICD laboratory and the other members in the VIRTUALIS project. The primary objective of the thesis is participating in the SP3-Global risk analysis, a sub-project of the VIRTUALIS project, to develop anomaly detection methods using data from sensors and cameras in a building. These methods could analyze this data to detect the suspect (terrorist, attackers) automatically. These techniques could be implemented in the real security system to assist the operators.

The VIRTUALIS dataset comes from a simulation model and a system of camera and other sensors which has been set up at Thales site. In the beginning, the consortium tended to apply the security system for two scenarios: the commercial building at Thales site and the Chatelet metro station. However, due to the unavailability of the dataset from metro station, we replace this scenario with the ATC dataset. The ATC dataset is a public dataset obtained by a system of 3D-cameras at the ATC shopping center in Osaka,

Japan. The purpose of this dataset is different from VIRTUALIS dataset as it tends to help people who get lost and need assistance in the building. A limitation exists in the VIRTUALIS real dataset, where the re-identification task was not done perfectly. So the methods that work with sequence can not use this dataset. With these datasets, we have developed two methods for the VIRTUALIS project. The first method *Abnormal Events Detection using key metrics* analyzes the trajectories as events in sub-area in the building. The second method *Abnormal Trajectory Detection* extracts trajectories into sequences of activities and builds a detectors for this type of sequence.

The AED technique is based on the parametric method for detection anomaly problem. This technique selects critical areas in the building and divide a day into multiple time windows. Then trajectories in these areas in each time window are transformed into key metrics. Then a multi-classifier which compute a threshold for each key metric in each time window is created, and these thresholds are used to detect abnormal key metrics. Another approach of the key metrics is constructing a single classifier using vector of key metrics in each time window as input. The one-class SVM classifier is then implemented to detect abnormal vector of key metrics. The experimentation shows an interesting result for the VIRTUALIS scenario where the multi-classifier can detect the attacking events in both simulated and real dataset and the single classifier gives considerable detection results. In the ATC dataset, because the labeled dataset is selected by experience of an expert, and there are many abnormal trajectories in the unlabeled dataset, this method does not perform well.

The ATD technique transforms the time series of movement points in trajectories into sequences of activities. In the VIRTUALIS dataset, they are simply inferred from the location. The activities are created efficiently using the office separations in the building. On the contrary, the ATC dataset is an open space, so the extracting stage is done by dividing the whole area into equal size sub-areas, and sequence of activities is considered as the sequence of sub-areas. After extracting the sequences, we create an unsupervised classifier One Class SVM. A specific dissimilarity function between sequences and kernels are defined to apply sequences for this classifier. Three types of kernel such as the GDTW, DTA, and GA are applied. The Kernel Alignment algorithm is used to search for the optimal hyperparameters. The experiments of the VIRTUALIS scenario show a great performance

for simulated dataset. In the ATC dataset, we made a test to compare our model with the author of this dataset and we obtained a good result. Then we recreate the test of the unsupervised one-class SVM model to compare the performance of different kernels.

These two techniques have their advantage and disadvantage. For a security system, they cover the weakness of each other. The AED technique is simple and works in a critical area, so only the sub-trajectory inside the area is supervised, and the ATD technique analyzes the whole trajectory. However, the ATD technique is limited by the re-identification problem, which means the whole trajectory of a person is interrupted and we can not have the whole trajectory in real dataset. The AED is not affected by this problem. Our best contribution in this thesis is developing these techniques for the VIRTUALIS project to improve the security system for critical infrastructure. These techniques are applied to two different scenarios.

6.2 Perspective

In the future, even though the VIRTUALIS project has finished, there still exist many problem that need to be solved. The first problem is improving the pedestrian detection, recognition and re-identification the pedestrian with high accuracy in a public place. This problem is difficult due to many reasons such as the unstable environment (light condition), too many people in the camera's field of view (metro station scenario). The second problem is the scale of the model when the size of the dataset is bigger. The One Class SVM is pretty bad with the scale problem, it takes long computation time for the gram matrix because of the custom dissimilarity function. For example if there are 1000 sequences in the training set, each sequence has an average of 5 activities, so the cost function is called $1000^2 * 5^2 = 25 * 10^6$ times. So the increasing amount of training sequence will cause more trouble to train the model.

For the method AED, the technique could be improved by specifying the model for each category such as human functionality, period. For example, if we have more days in the VIRTUALIS dataset, we can compute specific thresholds for each key metric on each period such as for each day in a week or special period in a year. If we can recognize the

identity of each person through the camera, we can compare their role (engineer, secretary, guest) in a reference list, and apply the specific model for them (a classifier for engineer, secretary).

There is also much room to improve the DTA method. First, starting from the pre-processing stage, there are other methods to represent trajectory such as tensor or matrix as introduced in [131]. With this representation, we can compute the similarity between every person in the building by using methods such as collaborative filtering, and therefore identify anomalies. The second improvement could be done to change the similarity measure. Until now this is a function with an arbitrary choice of hyperparameters. If we can define a better function which can compare the distance between two activities, the detection result will be improved.

Out of the scope of the proposed methods, due to a great trending of using Deep Neural Network to solve machine learning problems, we should try to apply Recurrent Neural Network, especially Long Short-Term Memory (LSTM) to solve our detection abnormal sequence problem. In fact, as our dataset is mainly trajectory, we can train a model to predict the next movement of each people, and if one's trajectory is very different from our prediction, it is classified as an anomaly.

Chapitre 7

La détection des événements atypiques pour la sécurité infrastructures critiques

7.1 Introduction générales

Cette thèse vise à développer des approches pour la détection d'anomalies dans les systèmes d'infrastructures critiques. Il a été réalisé dans le cadre du projet VIRTUALIS du programme FUI15. VIRTUALIS a réuni dix partenaires parmi lesquels l'équipe de modélisation et de fiabilité du système (M2S) du laboratoire de l'ICD. L'objectif de ce projet était de développer un système de sécurité automatique pour les infrastructures critiques. Notre contribution principale au projet était de proposer des méthodes de détection pour les applications de sécurité, en utilisant des données provenant de caméras et de capteurs. Le projet VIRTUALIS a duré trois ans et a débuté en juillet 2014, environ un an avant le début de la thèse. Le travail a principalement été fait sur la base du projet. Cependant, en raison d'un manque de données du projet VIRTUALIS, des données publiques d'un projet appelé ATC ont également été utilisées.

Cette introduction commence par une brève description de l'infrastructure critique

suivie d'une description des problèmes de sécurité actuels et de la motivation de VIRTUALIS. Une brève introduction du projet ATC est également mentionnée. Il se termine par une brève présentation de notre contribution principale au projet VIRTUALIS et des grandes lignes du manuscrit.

Les infrastructures critiques sont des systèmes complexes [30], et chaque composant de l'infrastructure constitue une petite partie du réseau complexe qui forme l'infrastructure globale. L'état d'une infrastructure est corrélé à l'état de l'autre, ce qui crée un réseau d'interdépendance entre les infrastructures dans un système étendu. Les infrastructures critiques couvrent les bâtiments publics tels que les aéroports, les gares et les stations de métro, les administrations, les bâtiments tertiaires (banques, bureaux) et les sites industriels (raffineries, centrales nucléaires, stations d'épuration). Les infrastructures critiques de la nation fournissent les services essentiels qui soutiennent la société et servent d'épine dorsale de l'économie, de la sécurité et de la santé de la nation. Ces infrastructures fournissent aux gens la puissance qui est utilisée dans les maisons, l'eau à boire, le transport, les magasins pour faire des achats, et les systèmes de communication.

Le goulot d'étranglement des mesures de sécurité est la tâche écrasante de l'homme (le travail de longue durée de l'opérateur), entraînent le manque de fiabilité du système de sécurité. Ainsi, un système de surveillance automatique pourrait être la contre-mesure particulière qui peut être appliquée à de nombreuses infrastructures critiques pour réduire les risques de sécurité. Par conséquent, le développement d'un tel système est critique et urgent dans la situation actuelle. En plus des nouvelles politiques dans le protocole de sécurité, le gouvernement français a investi dans de nombreux projets pour créer une feuille de route mondiale pour le système de sécurité d'auto-surveillance. L'objectif est de développer différentes briques technologiques pour la sécurité des infrastructures. Le projet VIRTUALIS est l'une des briques essentielles, qui est la motivation et fournit la ressource pour cette thèse.

Le projet VIRTUALIS

L'objectif du projet VIRTUALIS est de créer un système de sécurité automatique pour les infrastructures critiques telles que les bâtiments commerciaux, les stations de métro. Ce système peut surveiller les activités de chaque personne sur le site, sur la base des

informations recueillies par un grand nombre de capteurs, parmi lesquels cameras [31], puis analyse les données et alerte pour prévenir l'attaque avant qu'elle ne survienne. Un tel système peut aider les opérateurs à détecter les événements anormaux de manière plus efficace, augmentant ainsi l'efficacité d'un système de sécurité.

Le projet est divisé en plusieurs sous-projets, regroupés dans des paquets portant sur l'administration, le droit, les tâches techniques. Chaque sous-projet est géré par un consortium particulier. Le travail de recherche présenté dans le manuscrit a porté sur le sous-projet 3 - Analyse du risque global (SP3). C'est une étape intermédiaire, qui contient plusieurs modèles de détection développés par les membres du consortium SP3. Chaque modèle utilise des données pré-traitées provenant de plusieurs types de capteurs, préparés par les sous-projets précédents, et génère l'anomalie détectée. Une anomalie peut être un événement anormal ou quelque chose ou le comportement inhabituel de quelqu'un.

Par défaut, deux types d'infrastructures, une station d'échange de métro et un immeuble de bureaux ont été choisis comme scénarios pour le projet VIRTUALIS. Le site de Thales héberge le scénario du bureau de travail et la station de métro Chatelet à Paris accueille le scénario de la station de métro. En raison de ce plan, des systèmes de capteurs devaient être mis en place sur le site de Thales et sur la station de métro Chatelet, ce qui correspond à deux scénarios de collecte de données dans ces environnements. Ensuite, chaque membre a formé ses modèles de détection sur le même ensemble de données et ces modèles seraient intégrés dans un système de sécurité automatique. Le résultat du projet serait affiché à une technique de démonstration. Malheureusement, le processus d'acquisition des données de la station de métro s'est heurté à des problèmes administratifs, ce qui a conduit à l'abandon du scénario de la station de métro. En conséquence, lors de la démonstration finale du projet VIRTUALIS, le système proposé s'est déroulé dans le scénario du site de Thales.

Comme mentionné précédemment, le projet VIRTUALIS a été terminé un an avant la thèse. Ainsi, après le projet VIRTUALIS, on a encore le temps d'expérimenter nos méthodes dans un second cas. On a cherché un ensemble de données riche et robuste pouvant remplacer le cas de la station de métro. On a finalement retenu l'ensemble de données publiques du projet ATC et l'utilisons comme un deuxième cas expérimental.

Le projet ATC [19] a été réalisé au Japon, un système de détection a été installé dans le centre commercial ATC de la baie d'Osaka. Les informations détaillées du projet ATC seront présentées dans la section 7.2.2.

Aperçu de la thèse

La contribution principale de ce travail consiste en deux méthodes de détection d'anomalies utilisant des données de trajectoire. Les deux méthodes proposées sont complètement différentes, et elles peuvent couvrir la limitation de l'autre dans un système de sécurité. Ces méthodes s'appellent *La détection des événements atypiques en utilisant des métriques clés (DEA)* et *La détection des trajectoires atypiques (DTA)*. Les méthodes sont appliquées à l'affaire VIRTUALIS (immeuble de bureau) et à l'affaire ATC (centre commercial). Le problème pour chaque situation est différent, de sorte que les méthodes sont adaptées aux caractéristiques spécifiques de chaque environnement.

- DEA est une méthode paramétrique, basée sur des entités extraites. Cette méthode analyse plusieurs sous-trajectoires observées dans une zone critique du bâtiment. Il considère les sous-trajectoires comme des événements qui se produisent dans la zone et extrait les métriques clés de ces événements. Une étape de formation permet d'apprendre la distribution des métriques clés et de choisir les seuils de détection en fonction de la demande opérationnelle. Les seuils sont appliqués à l'étape de détection pour classer les situations anormales dans la zone.

- DTA est une méthode qui détecte les séquences anormales d'activités. Il traite les données pour produire des séquences d'activités, où chaque activité est déduite des données recueillies. La fonction de détection est formée en fonction des données disponibles. Selon l'ensemble de données disponibles, l'entraînement du classificateur pourrait être supervisé en utilisant des données normales et des données d'attaque, ou non supervisé en utilisant uniquement des données normales. Le classificateur est ensuite appliqué pour prédire l'étiquette de nouvelles séquences.

Ces méthodes ont été publiées sous la forme de deux articles ([69], [70]).

Le plan de la résumé en français

La section 7.2 décrit le projet derrière ce travail et les données utilisées dans ce manuscrit. La description du projet VIRTUALIS comprend le consortium, le système de capteurs, le développement de l'ensemble de données en relation avec un scénario anormal particulier. En plus de ces données, l'ensemble de données public du projet ATC sera présenté.

Pour fournir un contexte théorique pour nos développements scientifiques, la section 7.3 décrit la formulation du problème, qui est la détection d'anomalie pour les données de trajectoire, et catégorise les méthodes utilisées par d'autres chercheurs pour répondre à ce problème. On passe en revue certains travaux existants, identifions les principales lacunes de la recherche afin de positionner nos contributions.

La section 7.4 présente la première contribution nommée DEA. La section 7.5 présente la deuxième contribution nommée DTA. Dans les deux chapitres, la méthodologie est présentée, et la mise en œuvre des deux cas de test VIRTUALIS et ATC sont expliquées respectivement.

La conclusion et la perspective du développement futur des systèmes de sécurité automatiques sont présentées à la section 7.6.

7.2 Les projets et les données

Ce chapitre présente le projet VIRTUALIS et les contextes d'application dans lesquels on envisage de défier les méthodes proposées. L'un des points clés étant lié aux données, alors une attention particulière sera accordée à la présentation de chaque ensemble de données. Ce chapitre est organisé en deux parties principales. Le premier présente le projet VIRTUALIS. Dans le cas de VIRTUALIS, on a deux ensembles de données, un simulé et un réel. Des scénarios d'attaque ont été préparés par des experts en sécurité pour tester les méthodes proposées. La deuxième section présente le cas ATC. Et on discute également de la caractéristique du système de capteur et de l'environnement différent de chaque scénario.

7.2.1 Le projet VIRTUALIS

La description du projet

Le projet VIRTUALIS s'inscrit dans une démarche globale visant à proposer un système de gestion intégrale de la sécurité physique et logique d'une infrastructure critique, qui permette d'optimiser et de faciliter la gestion des risques pour l'installation et l'utilisation du système de sécurité. Ce système fournira des alertes et des procédures de sécurité vérifiées et validées.

Le consortium du projet VIRTUALIS provient de nombreuses entreprises et organisations dans différents domaines. Le laboratoire M2S sitúe à l'Université de Technologie de Troyes (UTT) participe principalement aux sous-projets SP3, et cette thèse ne contribue qu'à ce sous-projet. Dans cette tâche, chaque partenaire fournit une technique qui utilise les données de trajectoire des capteurs en entrée et génère l'alerte s'il détecte des choses d'anormales. En raison de la contrainte, UTT propose quatre techniques dans le projet dont deux techniques sont dans cette thèse.

Le sous-projet SP3 - l'analyse de risques

Le sous-projet SP3, piloté par SISPIA, a pour objectif de proposer différentes techniques qui analysent les données provenant de plusieurs types de sources tels que caméra, lecteur de carte, capteur d'iris, etc. La sortie de chaque technique est recueillie, et une fusion technique, dans la dernière étape, décidera si le système doit déclencher l'alarme ou non. Selon le calendrier du consortium dans SP3, la tâche SP3 est divisée en deux étapes, correspondant à deux scénarios : la centre commercial à Thales et la salle d'échange de métro. Cependant, dans la période de développement, seules les données du bâtiment Thales sont disponibles. Ainsi, le système de sécurité du projet VIRTUALIS ne fonctionne que sur le scénario de bureau chez Thales.

L'environnement

Deux cas d'usages sont prévu dans le projet VIRTUALIS : la station de métro et le bureau. La station de métro est un espace public multifonctionnel pour transporter de nombreuses personnes vers des lieux de travail, des stations ou des vacances. C'est une zone de mélange social, les passagers viennent d'origines diverses avec des destinations différentes dans la station. Contrairement à la station de métro, les bureaux sont privés et fermés au public. Au-delà du risque terroriste, les principaux risques que les entreprises cherchent à réduire sont des actes de sabotage et d'espionnage qui mettent en péril les actifs de l'entreprise. Par conséquent, toute personne présente dans l'infrastructure ou connectée au système d'information doit être autorisée. Ces bâtiments tertiaires sont pour la plupart équipés de systèmes de sécurité assurant le contrôle d'accès, la protection périmétrique, la détection incendie et le lecteur de carte au système d'information.

Le site Thales à Palaiseau accueille Thales Research & Technology France comme principal occupant. Ceci est un exemple du contexte de bureau. Plusieurs types d'installations techniques sont présents sur le site : laboratoires, bureaux et autres zones dans un état opérationnel de soutien aux objectifs du Centre de Recherche. Les expériences du scénario de bureau dans le projet VIRTUALIS ont été exécutées à cet endroit.

Le système de capteurs composé de caméra et de capteurs a été installé dans la zone ThereSIS comme sur la Figure 2-4. Plusieurs types de capteurs tels qu'un lecteur de cartes, capteur d'iris, détecteur de mouvement (Figure 2-4a) contrôlent chaque activité dans la zone ThereSIS. Chaque employé doit scanner sa carte chaque fois qu'il entre et sort de la zone (entrée A3 et A4). Un autre contrôle de lecteur de carte est placé à la porte du serveur et à l'imprimante, et seuls les employés autorisés sont autorisés à accéder à la salle des serveurs ou à utiliser l'imprimante. D'autres capteurs ont été installés pour détecter des événements anormaux dans la zone. La distribution des caméras est présentée dans la figure 2-4b, et les champs de vision de la caméra étaient concentrées sur la zone de l'imprimante (Figure 2-5).

Depuis le début du projet VIRTUALIS, tout le sous-projet a démarré en même temps, alors que le SP2 (le sous-projet avant SP3 qui fait l'installation de système des capteurs) a rencontré de nombreux problèmes lors de l'installation du système de capteurs

chez ThereSIS pour collecter des données réelles, ou en accord sur le format des données, les méthodes proposées de SP3 ont été développées par le consortium SP3 et ont nécessité des données pour former et valider les modèles. Par conséquent, un ensemble de données temporaire est requis pour développer des méthodes dans SP3, c'est la motivation pour créer l'ensemble de données simulé.

Le donnée simulée

L'ensemble de données simulé a été créé par les services Thales SAS. Tout d'abord, le bâtiment simulé a le même plan que le bâtiment réel, puis les gens simulés sont ajoutés à la simulation.

L'ensemble de données simulé et l'ensemble de données réel sont composés de deux parties : un ensemble de données environnementales sur la position du bureau, des éléments dans le bâtiment et un ensemble de données opérationnelles contenant les trajectoires de la caméra et les observations des capteurs. L'ensemble de données environnementales simulé contient le CdV de la caméra et la coordonnée des éléments critiques. La zone couverte par la caméra est en polygones verts comme dans la figure 2-6.

La structure de données de la trajectoire est composée de 4 caractéristiques : Temps, Coordonnée X, Coordonnée Y, Personne ID, correspondent à chaque observation à chaque horodatage pour chaque personne. La table 7.1 donne un exemple de trajectoire d'ID 101.

TABLE 7.1 – Le donnée simulé brut.

ID	Time	X	Y
101	9 :20 :01	4.5	-5
101	9 :20 :02	4.6	-4.8
101	9 :20 :03	4.7	-4.6
101	9 :20 :04	4.8	-4.3

Il y a des informations supplémentaires telles que le journal des lecteurs de cartes à de nombreux endroits comme l'entrée, l'imprimante. Cet ensemble de données on donne plus d'informations sur ce qui se passe dans le bâtiment et il permet de tester de nombreuses méthodes pour analyser et détecter des événements anormaux.

Le donnée réel

Deux acquisitions de données ont été organisées dans les locaux de Thales Research & Technology à Palaiseau, dans la zone ThereSIS du bâtiment. La première acquisition s'est achevée du 22 au 26 février 2016 et la deuxième acquisition a eu lieu du 7 au 10 novembre 2016. Dans l'ensemble de données environnementales réelles, il existe différentes informations disponibles :

- **Personnel Identifiant** : identifiant unique enregistré pour chaque salarié sur le site Thales.
- **Position de l'élément** : coordonnée de chaque élément à l'intérieur du bâtiment, depuis les capteurs, les portes, l'ascenseur jusqu'à l'imprimante, etc.
- **La zone de la surveillance** : le champ de vision de la caméra dans le bâtiment. Cependant, le champ de vision ne couvre pas toutes les zones du bâtiment telles que l'intérieur du bureau, et certains coins "morts" dans le couloir. La zone de l'imprimante est couverte dans de nombreux champs de vision de la caméra.

Le donnée opérationnel réel détient les enregistrements de chaque activité dans le bâtiment. C'est un énorme ensemble de données avec une quantité croissante de données. Cet ensemble de données comprend les données brutes de capteurs et de données après l'étape de pré-traitement appelée données de visualisation. Ces jeux de données ont été créés par les membres du SP2. Puisqu'on travaille sur le sous-projet SP3, on s'intéresse uniquement aux données de virtualisation et nos méthodes sont développées à partir de cet ensemble de données. Il existe quatre catégories de données de visualisation, elles sont présentées comme suit :

- **Contrôle d'accès** : chaque observation dans cette catégorie est créée après une connexion par un lecteur de carte ou un capteur Iris. Cette catégorie contient la sortie de la reconnaissance Iris, double iris/carte d'authentification, double visage/carte d'identification.

- **Localisation** : l'information de localisation est la plus grande catégorie. Il contient les résultats de la vérification de l'iris et les trajectoires capturées de toutes les caméras. Cette catégorie contient la sortie du module de fusion multiple-caméra, suivi visuel, tâche de reconnaissance du visage et de l'iris.

- **Corrélation** : la catégorie de corrélation fournit l'information pour connecter des événements discrets de localisation. Il connecte deux instances de type de localisation éventuellement liées à la même personne. C'est l'information la plus importante pour construire une trajectoire complète. Il contient la sortie de plusieurs modules du sous-projet SP2, comprend le module de fusion multiple-caméras, la reconnaissance du visage et de l'iris, le double iris/carte d'authentification, et le visage/carte.

- **Intrusion** : la catégorie d'intrusion regroupe les informations issues du capteur de porte infrarouge (détection de présence non visuelle), qui capture chaque événement lié à une entrée et une sortie pour les portes de la salle des serveurs et du bâtiment.

Le problème de ré-identification est l'un des problèmes les plus compliqués dans un système de surveillance. La ré-identification implique qu'une personne qui a déjà été vue est identifiée lors de sa prochaine apparition dans un système de caméras. La difficulté de cette tâche provient du changement d'apparence sur les vues de la caméra en raison de la grande différence d'éclairage, de point de vue ou de caractéristiques de la caméra. La plupart des approches actuelles s'appuient sur la similarité basée sur l'apparence entre les images pour établir des correspondances ([12], [27]) mais le résultat est encore limité. C'est une limite qu'on a rencontrée dans ce projet, ce qui pose des problèmes pour certaines des méthodes proposées qui analysent les trajectoires.

La scénario d'attaque

Avec le système des capteurs dans le bâtiment, les données sont collectées pour former nos méthodes proposées. Ces données contiennent les activités normales de tous les habitants du bâtiment Thalès. Cependant, les méthodes proposées ne peuvent pas être testées avec seulement des activités normales dans l'ensemble de données. Par conséquent, des scénarios d'attaque ont été préparés par des experts en sécurité pour tester et évaluer

les performances des méthodes proposées dans le sous-projet SP3 pour la tâche de détection anormale.

Ces scénarios simulent des attaques de concurrents ou d'ennemis qui veulent voler des informations confidentielles de la base de données de Thalès. Les attaquants pouvaient profiter des équipements connectés au réseau professionnel (Figure 2-3) comme les imprimantes, les téléphones ou les PC pour accéder au serveur. Les scénarios d'attaque ont donc été créés, en se plaçant du point de vue de l'attaquant, en essayant d'accéder aux éléments critiques tels que l'imprimante et la salle des serveurs. Ces scénarios d'attaque ont été construits pour des jeux de données simulés et réels avec une petite différence entre eux.

7.2.2 Le cas ATC

La description du projet

Cet ensemble de données provient du projet ATC qui s'est déroulé au centre commercial ATC à Osaka, au Japon. Cet ensemble de données est volumineux car les auteurs ont mis en place un système de capteurs pour collecter des données sur 92 jours répartis sur une année d'octobre 2012 à novembre 2013. Dans le cadre d'un projet visant à permettre aux robots sociaux mobiles de fonctionner dans les espaces publics (page d'accueil du projet en japonais, fondée par JST / CREST), un environnement de suivi est mis en place dans le centre commercial «ATC». Le système se compose de plusieurs capteurs de distance 3D, couvrant une superficie d'environ $900m^2$.

Le centre commercial ATC est considéré comme similaire à la station de métro en raison de sa source compliquée de visiteurs et de sa vaste zone de surveillance. Contrairement au bureau de Thales, où les caméras ne sont installées que dans le couloir et où la CdV de caméra est limitée à l'intérieur du bureau, la majeure partie de l'ATC est couverte par le système de la caméra et tous les mouvements des piétons dans le bâtiment sont capturé à partir du moment où ils entrent jusqu'à ce qu'ils partent. Bien que l'objectif de cet ensemble de données ne soit pas la sécurité, la quantité et la qualité des trajectoires dans cet ensemble de données attirent toujours notre attention, et on a décidé d'utiliser cet ensemble de données pour l'analyse dans la deuxième étape de la thèse.

Dans le projet ATC, les auteurs ont expliqué la raison pour laquelle ils ont choisi les données de gamme sur les données RVB de la caméra [19]. Trois types de capteurs de distance 3D sont choisis pour l'expérience : Temps-de-Vol cameras, Lumière Structuré cameras, Multi-layer laser scanners.

L'environnement

Le plan de configuration de trois types de capteurs ci-dessus dans le site ATC est présenté dans la figure 2-7. Ce bâtiment est un endroit bondé, il relie la gare, les bâtiments commerciaux et un quai. Donc, le piéton dans ce bâtiment a des objectifs et une destination différents. En général, le bâtiment est divisé en un carré central et un couloir. La place centrale est l'endroit où les gens restent pour attendre le train ou le bateau, c'est aussi un lieu d'exposition. D'un autre côté, le couloir est plus étroit et les gens marchent principalement dans cette zone.

Le donnée réel

En général, la collecte des données a été effectuée chaque semaine le mercredi et le dimanche, l'objectif est de prendre un échantillon de jour de travail et un jour de week-end, pour limiter le volume de l'ensemble de données mais conserver l'information. L'ensemble de données est fourni sous forme de fichier CSV, chaque ligne correspond à une seule observation d'une personne dans un instant. Les caractéristiques suivantes sont disponibles :

Table 7.2 – Les trajectoires brut de donnée ATC.

Fonctionnalité	Description	Exemple
Temps [ms]	L'instant de l'observation	1353202633.801
ID personne	Numéro d'identité attribué à chaque trajectoire	10370300
X, Y coordination [mm]	Coordonnée de la personne dans un espace prédéfini	-326 2054
Z coordination [mm]	Hauteur de la personne	1622
Vélocité [mm/s]	La vitesse de la personne à l'instant	912.316
Angle de mouvement [rad]	L'angle de déplacement de la personne par rapport au l'axe d'origine	-1.375
Angle de face [rad]	L'angle entre le visage et l'épaule	0.548

L'ensemble de données ATC a plus de fonctionnalités que l'ensemble de données VIRTUALIS. La hauteur du piéton est estimée afin qu'on puisse prédire si cette personne est un enfant ou une personne mûre. La vitesse à chaque observation est également conservée. L'*angle de mouvement* permet de comparer l'angle de déplacement de la personne à un axe prédéfini. En outre, *l'angle de face* est l'angle entre le visage et le corps, il peut dire si une personne regarde autour de soi ou va droit. Il existe un petit jeu de données étiqueté qui a été créé pour une expérience dans [18]. Cependant, la signification de l'anomalie est une personne qui semble se perdre et avoir besoin d'aide.

7.3 La détection des anomalies pour les trajectoires

La découverte de trajectoires anormales dans un ensemble de données à grande échelle a également une signification et une valeur pratique remarquables. Il est directement concernée par cette thèse. Ce chapitre vise à présenter les méthodes et l'application de la détection d'anomalie principalement appliquées aux trajectoires. La premier partie parle de la définition générale du problème de détection d'anomalies et de quelques méthodes populaires pour la détection d'anomalies. On catégorie ces méthodes en deux catégories : basées sur la densité et basées sur la frontière. Dans la deuxième partie, on approfondisse la définition précise d'une trajectoire ainsi que les méthodes pour traiter ce type de données. Ces approches on a motivés à proposer nos méthodes pour le projet VIRTUALIS.

7.3.1 L'enquête

La détection des anomalies est largement utilisée dans diverses applications. De nombreuses méthodes ont été proposées pour résoudre ce problème. L'approche générale de la méthode de détection d'anomalies consiste à définir une frontière entre les instances normales et les instances anormales en cas d'apprentissage supervisé et une limite autour des instances non étiqueté (la majorité est normale) en cas d'apprentissage non supervisé. Ensuite, cette limite est utilisée pour donner l'étiquette pour les nouvelles instances qui n'appartiennent pas aux régions normales comme des anomalies. Le tableau 7.3 contient les enquêtes complet sur cette domaine.

TABLE 7.3 – L'enquête des méthodes pour la détection des anomalies.

Author	Paper	Summary
Chandola et al.	[25]	Ce document fournit un aperçu structuré et complet de la recherche sur la détection des anomalies. Il comprend la définition, les défis, les travaux connexes, les différentes phases du problème de détection d'anomalies, les applications ; plusieurs types de techniques. En bref, tout sur la détection d'anomalie.
Agrawal et al.	[3]	Cet article passe en revue diverses techniques d'exploration de données pour la détection d'anomalies afin de mieux comprendre les techniques existantes qui pourraient aider les chercheurs intéressés à travailler dans cette direction.
Parmar et al.	[86]	Ce travail se concentre sur la détection des anomalies dans l'exploration de données. L'objectif principal est de détecter l'anomalie dans les données de séries temporelles en utilisant des techniques d'apprentissage automatique.

Dans cette thèse, on sépare les techniques d'anomalies de détection en deux catégories principales : basées sur la densité, basées sur la frontìere. Les techniques basées sur la densité utilisent la distribution des données et les techniques basées sur la frontìere appliquent les instances qui se trouvent sur le bord de la distribution de données. Les techniques de ces groupes sont présentées dans les sections suivantes. Deux éléments essentiels pour ces techniques sont la fonction de similarité et les fonctions du noyau sont également discutées.

7.3.2 La méthode

Les techniques basées sur la densité

Une méthode basée sur la densité utilise la distribution de l'ensemble de données d'apprentissage pour créer la limite de détection. En général, dans l'espace de donnée, la zone qui a une densité élevée de points de données a plus de chance d'être dans la classe typique que la zone à faible densité ou sans point de données. Le groupe des techniques basées sur la densité contient les techniques de regroupement et les techniques paramétriques. Les techniques de regroupement contiennent trois principal approches : DBSCAN ([41], [56]), k-Means ([80], [95]). Les techniques paramétriques contiennent les modeles Gaussiennes et les modeles des distributions mixe ([2], [25]).

Les techniques basées sur la frontière

Les méthodes basées sur la frontìere construisent la limite de détection en fonction des points de données situés sur la limite de la classe de données. Les principales méthodes représentant ce groupe sont SVM ([46], [55], [67], [120], [20]) pour la classification supervisée et seul-classe SVM ([42], [72], [107], [26]) pour la classification non supervisée. SVM est un classificateur linéaire qui trouve la limite qui maximise la marge entre deux classes de données séparables. Le méthode seul-classe SVM trouve un hyperplan dans un espace de caractéristiques $\mathbb{H}$, qui maximise la distance des données d'entraînement depuis l'origine dans cet espace, alors qu'une petite fraction des données se situe entre l'hyperplan et l'origine.

Les mesures des similarités

Dans le problème d'anomalie de détection, l'une des tâches les plus cruciales consiste à comparer deux instances de l'ensemble de données. Les mesures de dissimilarité populaire a été mentionnée dans [129]. En raison du type différent de données disponibles, on divise la mesure de similarité en deux groupes : la mesure de dissimilarité pour les points et pour les séquences. Le plus populaire méthode dans la dissimilarité pour les points est la distance Euclidien (l_2-norm), et le DTW est la méthode principale utilisé pour comparer deux séquences. L'algorithme DTW est appliqué dans plusieurs d'applications, telles que la classification ([89], [50]) ou la clustering ([59], [57]).

Les noyaux

La méthode du noyau est devenue un outil populaire pour les techniques de détection d'anomalies [105]. Fondamentalement, il est associé à SVM et est indispensable à seul-classe SVM pour résoudre le problème de classification non-linéaire [33]. L'idée clé de la méthode noyau est que de nombreuses techniques d'apprentissage automatique comme SVM ou K-means utilisent le produit scalaire entre les vecteurs lors de la création d'un modèle de données, et que ce point peut être remplacé par un notion mathématique plus générale appelée noyau.

De nombreux types de noyaux ont été développés, mais on peut les classer en deux groupes correspondant aux points et aux séquences. Les noyaux de point sont des noyaux ordinaire tel que le noyau polynomial et le noyau Gaussienne. Pour les séquences, il y a trois types de noyau qui sont considère dans notre travail. Le premier noyau que je veux introduire est le noyau GDTW. Ce noyau a été utilisé dans de nombreuses applications pour les données de séries temporelles ([4], [130], [54]). Le deuxième noyau pour la séquence est le noyau Dynamic Time Alignment, qui a été introduit dans [10] et appliqué dans [111] pour le problème de classification SVM et qui détecte les mouvements faciaux dans [133]. Le troisième noyau est le noyau GA, qui a été présenté par Cuturi et al ([36], [35]).

Donc jusqu'à ce point, on a discuté des techniques de détection d'anomalie pour les données de point et de séquence. La partie suivante de ce chapitre présente les problèmes de détection d'anomalie liés aux données de trajectoire, qui sont un type de séquence. Il donne un aperçu de la façon dont d'autres chercheurs traitent ce type de données dans certaines applications.

7.3.3 L'issue avec les trajectoires

Le développement de nouvelles technologies a entraîné une augmentation massive du nombre de dispositifs sensibles à la localisation au cours des dernières années, ce qui a permis d'obtenir un nombre énorme de données sur les trajectoires. Les sources de données de trajectoire sont très diverses : dispositif de positionnement global (GPS) [118], [87], système global de communications mobiles (GSM), identification par radio-fréquence (RFDI) [134], Wifi [112] et le caméra [110], [21]. En étudiant la façon dont les autres chercheurs traitent les données de trajectoire, on reconnaisse qu'il existe deux approches principales : la détection d'événements anormaux et la détection de trajectoires anormales.

La première catégorie détecte les anomalies dans une sous-zone en utilisant de nombreuses sous-trajectoires dans un laps de temps. Les anomalies de cette catégorie sont causées par la présence d'une ou plusieurs trajectoires anormales combinées. Les chercheurs ont tendance à diviser l'ensemble de l'environnement en plusieurs petites sous-zones, alors seule la sous-trajectoire qui se trouve à l'intérieur de la sous-zone critique est considérée ([74],[84],[85],[16]).

La deuxième approche pour la détection d'anomalies à l'aide de données de trajectoire analyse l'ensemble de la trajectoire. Cependant, la trajectoire du point de mouvement est très difficile à classifier une trajectoire anormale par rapport aux trajectoires normales car la position précise de chacun est différente, et les modèles qui analysent ce type de trajectoire pourraient faire face à un taux élevé de fausses alarmes. La représentation graphique de la trajectoire est l'une des méthodes les plus populaires pour représenter la trajectoire, où l'environnement (ville, bâtiment) est divisé en plusieurs sous-zones (nœuds) et connexions (bords). De nombreux chercheurs ont utilisé les termes POI et AP pour représenter les nœuds et les bords respectivement et les trajectoires sont extrait en des séquences de POI et de AP ([79],[78], [83], [28]).

Sur la base de ce point de vue, on a proposé deux techniques : DEA et DTA, pour détecter les anomalies dans les infrastructures critiques. La première technique est une méthode paramétrique, qui traite les trajectoires comme des événements dans des sous-zones importantes du bâtiment. Le section 7.4 discutera de cette approche. La deuxième technique (DTA) du section 7.5 traite les séquences des POIs. Les deux techniques sont appliquées aux deux scénarios (bâtiment commercial et centre commercial). En raison des caractéristiques différentes de ces scénarios, chaque technique est modifiée pour s'adapter au contexte spécifique.

7.4 La détection des événements atypiques

7.4.1 L'introduction

Ce chapitre présente la méthode de détection des événements anormaux (DEA), la première méthode proposée pour le projet VIRTUALIS. Cette technique est motivée par l'idée de construire une méthode de détection pour protéger une zone locale à l'intérieur du bâtiment Après l'introduction de la méthodologie, on va plus loin dans l'explication de la façon dont on a sélectionné les zones critiques, les métriques clés, comment les calculer, et comment former les classificateurs. Ensuite, on discute de la mise en œuvre des deux scénarios : le bureau de travail et le centre commercial donné par les jeux de données VIRTUALIS et ATC respectivement.

7.4.2 La méthodologie

Cette méthode est composée de trois étapes : formation, validation et étape de détection.

Au cours de la formation, la première étape consiste à prédéfinir les paramètres des fenêtres temporelles et à sélectionner les zones critiques. A chaque fenêtre temporelle, on collecte tous les points d'une trajectoire à l'intérieur de la zone choisie (sous-trajectoire). Ces sous-trajectoires sont également appelées événements, ce qui pourrait être compris comme la présence de personnes dans une zone. Les événements dans la même fenêtre temporelle sont utilisés pour calculer les métriques clés, qui sont des caractéristiques caractérisant les activités des personnes pouvant être utilisées pour détecter un comportement anormal. Les types de métriques clés utilisées dans les expériences sont par exemple *durée moyenne d'une visite*, *nombre de visites*. Le vecteur qui contient les métriques clés de la même fenêtre temporelle est appelé vecteur des métriques clés. Ensuite, un classificateur est formé en utilisant les vecteurs de métriques clés de plusieurs jours dans l'ensemble de données. Il existe deux classificateurs proposés : l'ensemble classificateur et la seul-classe SVM classificateur. La résumé de cette étape est comme suivant :

1. Définir les éléments essentiels tels que les zones critiques, les paramètres de la fenêtre de temps ou les métriques clés.
2. Assigne les sous-trajectoires brutes dans la zone critique du jeu d'entraînement à chaque fenêtre temporelle.
3. Calculer les métriques clés en utilisant les sous-trajectoires dans chaque fenêtre temporelle pour créer des vecteurs de métriques clés.
4. Former un ou plusieurs classificateurs en utilisant les vecteurs de métriques clés.

L'étape de validation teste le classificateur formé en mode hors ligne. L'ensemble de validation est un ensemble de données étiqueté, qui comprend des événements décrivant des situations normales et des scénarios d'attaque. Cette étape permet de calibrer les paramètres du classificateur. Les mesures d'évaluation utilisées sont le taux réel vrai et le taux

négatif faux. La précision n'est pas utilisée car seuls quelques échantillons anormaux sont disponibles dans l'ensemble d'apprentissage.

L'étape de détection est effectuée en ligne, où un classificateur est déployé pour détecter les événements anormaux en temps réel.

L'étape de détection est basée sur le principe suivant : les événements sont observés en temps réel, le nouveau vecteur de métriques clés est calculé et ces vecteurs sont classés en utilisant les classificateurs formés. Le processus de détection est le suivant :

1. Collecte continuellement la trajectoire des caméras et des capteurs.

2. Sélectionnez uniquement les sous-trajectoires dans la zone prédéfinie.

3. Calcule le vecteur des métriques clés dans la fenêtre temporelle actuelle.

4. Prédire l'étiquette d'un nouveau vecteur de métriques clés en utilisant le classificateur.

5. Si la valeur observée est détectée comme anormale, une alarme est définie.

La division des zones

Cette méthode nécessite de définir certaines zones critiques de l'environnement. Ces zones peuvent être obtenues en divisant l'endroit en utilisant sa structure, par exemple, chaque pièce à l'intérieur d'un bâtiment est une zone. Dans un bâtiment, une zone critique peut être définie comme la zone autour d'un élément vital (un élément essentiel et qui attire les gens) comme une imprimante, un tableau électrique, un distributeur de billets dans une gare ferroviaire, etc. Une autre possibilité consiste à : en utilisant les caractéristiques spécifiques des piétons tels que les zones où les gens se déplacent et les zones où les gens restent. La segmentation de zone permet de définir une liste de zones. Puisque seules les zones critiques sont choisies pour appliquer la méthode, une sélection est basée sur la vulnérabilité et la criticité de la zone.

La fenêtre du temps

Par définition, les métriques clés sont des caractéristiques qui décrivent la caractéristique des visites dans une zone ou une caractéristique d'une zone dans une période. Un métrique clé du matin peut devenir anormal le soir, de sorte que les métriques clés sont calculées pour différentes périodes d'une journée. La fenêtre temporelle est utilisée pour cette tâche, elle a deux hyperparamètres : la durée et le décalage. La fenêtre temporelle peut se chevaucher si le décalage est inférieur à la durée et vice versa.

Les vecteurs des métriques clés

Dans cette technique, on intéresse à l'examen du comportement des personnes concernant leur présence dans une zone critique, on définisse donc des métriques clés pour décrire la situation d'une zone dans une fenêtre temporelle. Les valeurs des métriques clés sont organisées en vecteurs de métriques clés déterminées ci-dessous :

Notation 7.4.1. *Le vecteur des métriques clés X_t, $X_t \in R^{K \times D}$ associé à la fenêtre temporelle t est un vecteur bidimensionnel, où chaque ligne correspond à un jour spécifique dans l'ensemble de données et chaque colonne correspond à une métrique clé particulière. Chaque élément de ce vecteur est $m^k_{t,d}$, correspondant à une valeur observée de la métrique clé k, à la fenêtre temporelle t au jour d, $k \in [1 : K]$, $d \in [1 : D]$ où D est le nombre de jours et K est le nombre de métriques clés.*

Les principaux types de métriques clés que on propose sont *durée moyenne du séjour*, *durée maximale du séjour*, *nombre de visites* et *taux d'occupation*, qui peuvent varier dans trois domaines : type de personnes, lieu et du temps. Cela signifie qu'avec ces types de métriques clés, des métriques clés spécifiques sont créés pour différents types de personnes, de périodes ou d'emplacements en fonction des données d'entrée.

Ensemble classificateur

Ce classificateur crée une alerte pour chaque métrique clé. Il est construit en calculant un seuil pour chaque métrique clé dans chaque fenêtre temporelle.

Notation 7.4.2. *La variable aléatoire M_t^k est une métrique clé associé à une fenêtre du temps t qui a une valeur par jour : $M_t^k = [m_{t,1}^k, m_{t,2}^k, ..., m_{t,d}^k, ..., m_{t,D}^k]$, où K est le nombre de métriques clés et D est le nombre de jours dans l'ensemble de données, $k \in [1 : K]$, $d \in [1 : D]$.*

Les données utilisées dans ce classificateur sont les réalisations de la variable M_t^k. Et l'hypothèse suivante est faite : La métrique clé M_t^k associée à une fenêtre temporelle t est une variable aléatoire supposée correspondre à une distribution normale avec la moyenne $\mu_{M_t^k}$ et l'écart-type $\sigma_{M_t^k}$: $M_t^k \sim N(\mu_{M_t^k}, \sigma^2_{M_t^k})$. Dans la phase d'entraînement, la moyenne $\mu_{\hat{M_t^k}}$ et l'écart type $\sigma_{\hat{M_t^k}}$ de la métrique clé M^k sont estimé en utilisant les données de l'ensemble de donnée dans D jours.

Dans la phase de détection et de validation, dans une fenêtre temporelle spécifique t, la décision finale de déclencher l'alarme ou non est prise par plusieurs tests. chaque test a l'objective à décider si la valeur observé $m_{t,d}^k$ dépasse la seuil d'un métrique clé TM_t^k : $TM_t^k = \mu_{\hat{M_t^k}} + \alpha * \sigma_{\hat{M_t^k}}$.

Le seul-class SVM classificateur

Ce classificateur est proposé comme approche alternative pour analyser conjointement les métriques clés au lieu de les analyser séparément. Et les données utilisées dans ce classificateur sont un point de grande dimension combiné à partir des valeurs de métrique clés au même jour et à la même heure de plusieurs métriques clés. En particulier, chaque échantillon de ce point est une ligne dans le vecteur de la métrique clé X_t. Donc le format de donnée utilisé pour ce classificateur est le vecteur $v_{t,d} = [t, m_{t,d}^1, m_{t,d}^2, ..., m_{t,d}^k, ..., m_{t,d}^K]$.

Comme chaque instance est un vecteur numérique, on peut utiliser un noyau normal tel que le noyau RBF et la l_2-norm distance pour calculer la matrice du noyau pour le classificateur seul-classe SVM. Ensuite, on forme le modèle seul-classe SVM avec ce noyau, et ce modèle est utilisé pour prédire l'étiquette des nouvelles instances. En fait, ce classificateur est capable de détecter une valeur aberrante du vecteur de métriques clés, ce qui signifie que quelque chose d'anormal se produit dans la sous-zone prédéfinie. Il ne précise pas l'individu précis qui a causé cet événement anormal.

7.4.3 Le cas d'usage VIRTUALIS

Comme décrit ci-dessus, les métriques clés prévoient un rôle crucial pour classer les événements anormaux. Dans ce scénario, les métriques clés choisis pour les expériences sont les suivants :

- *Nombre de visites* : le but est de contrôler le nombre de personnes dans une zone sélectionnée.
- *Taux d'occupation* : le but est de contrôler le pourcentage de la fenêtre temporelle de sorte qu'au moins une personne occupe la zone.
- *Durée moyenne d'une visite* : cette métrique clé contrôle la durée moyenne du séjour.
- *Durée maximale d'une visite* : cette métrique clé contrôle la durée maximale de toutes les visites dans une fenêtre temporelle.
- *Nombre d'impressions* : contrôle le nombre de commandes d'impression dans la fenêtre de temps.

Les expérimentes

Cette section présente les expériences effectuées pour l'ensemble de données VIRTUALIS. Les expériences créent l'ensemble classificateur en utilisant les données des jours réguliers dans l'ensemble d'apprentissage. Ensuite, l'ensemble classificateur est utilisé pour détecter les événements anormaux dans la journée qui contient le scénario d'attaque. Les expériences sont réalisées avec un jeu de données simulé et un jeu de données réel.

La figure 7-1 montre le résultat de la détection dans la salle des serveurs. Les trois métriques clés *la durée moyenne*, *le nombre de visites* et *la ratio d'occupation* détectent l'accès anormal à la salle des serveurs entre 18 et 19 heures.

La figure 7-2 présente la courbe ROC du classificateur unique que on a implémenté pour le scénario VIRTUALIS. Le classificateur a été formé en utilisant un vecteur non étiqueté de métriques clés, composé de *durée moyenne dans une fenêtre temporelle*, *nombre*

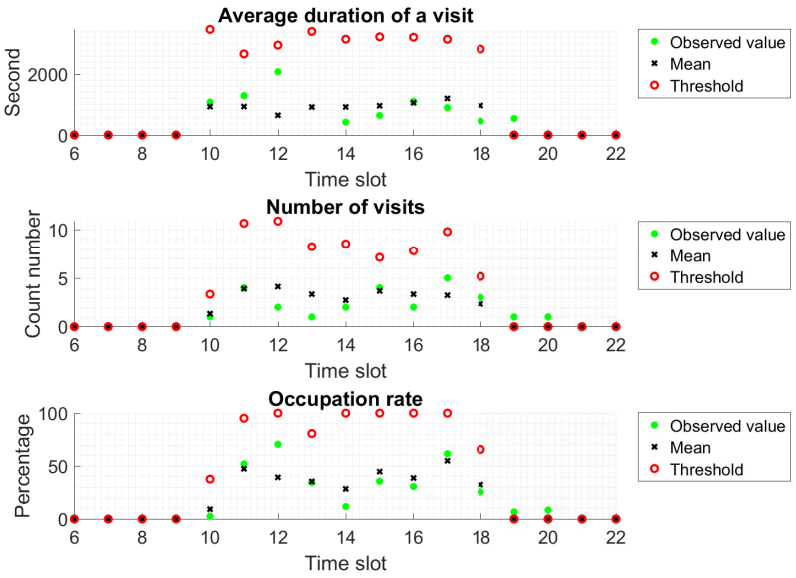


FIGURE 7-1 – La détection dans la salle serveur.

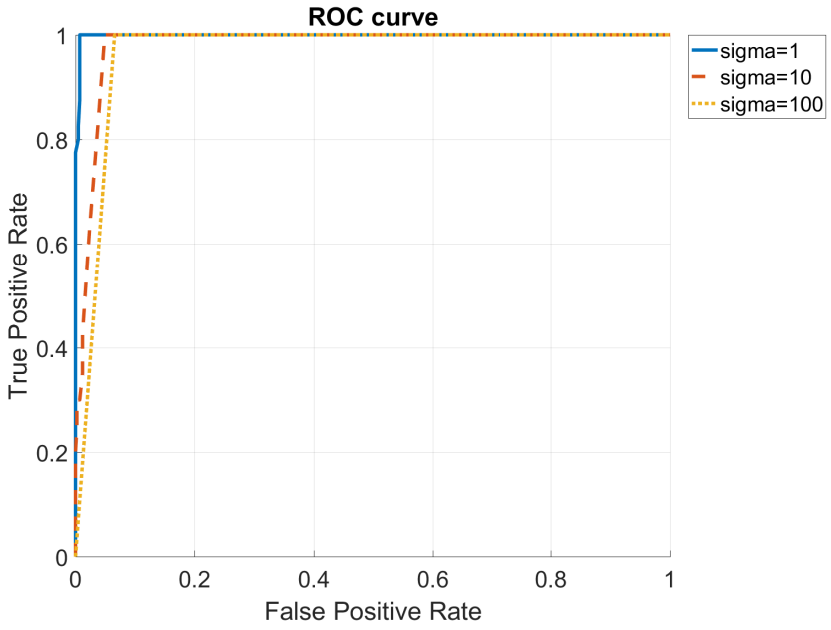


FIGURE 7-2 – Le courbe ROC de la détection dans la salle serveur.

de visites dans une fenêtre temporelle, *taux d'occupation* et la fenêtre. Le vecteur du jour d'attaque est utilisé pour le test, et la fenêtre temporelle qui contient les attaquants est considérée comme anormale. Cette figure montre une grande performance du classificateur

avec $\sigma = 10$, où le taux de détection est de 90 % et le taux de fausses alarmes est très faible (2 %).

7.4.4 Le cas d'usage ATC

Dans ce scénario, il n'y a pas de division naturelle comme les murs, et pas d'élément critique non plus. Selon le plan du centre commercial ATC, deux régions pourraient être détectées facilement. Ils sont le carré et le couloir comme sur la figure 2-7. On divise donc l'ensemble du bâtiment en deux parties après sa caractéristique différente : le carré et le couloir. Les gens ont tendance à rester sur la place pour attendre leur train ou leur bateau, et parfois des expositions sont organisées dans cette zone. Et le couloir est l'endroit où les gens ont tendance à marcher plus que rester car il relie l'immeuble de bureaux et la gare. La tâche de suivi dans ce scénario est beaucoup mieux, de sorte que les piétons sont suivis avec une grande précision. De plus, il n'y a pas d'éléments vulnérables sur lesquels se concentrer. La technique DEA pour l'ensemble de données ATC sera donc appliquée pour les zones principales ou l'ensemble du bâtiment.

- *Nombre de visites* : nombre de trajectoires dans une zone.
- *Taux d'occupation* : Le pourcentage de fenêtre temporelle que la zone prédéfinie est occupée par au moins un piéton.
- *Durée moyenne* : durée moyenne d'une visite.
- *Taille moyenne* : La hauteur estimée de chaque personne, ceci est obtenu en détectant le sommet de la tête des gens.
- *Vitesse moyenne* : la vitesse actuelle de chaque piéton.
- *Variance de l'angle de mouvement* : variance de l'angle du mouvement. Dans l'ensemble de données brutes, on a l'angle caractéristique du mouvement, qui décrit l'angle du corps à chaque instant. Cette métrique clé calcule la variation de cette fonctionnalité dans la zone prédéfinie. L'objectif est de détecter l'incohérence du piéton dans le mouvement.

- *Variance de l'angle de visage* : variance de l'angle entre le visage et le corps. Similaire à la variance de l'angle de mouvement, cette fonction tente de détecter si un piéton tourne beaucoup la tête en mouvement.

Les expérimentes

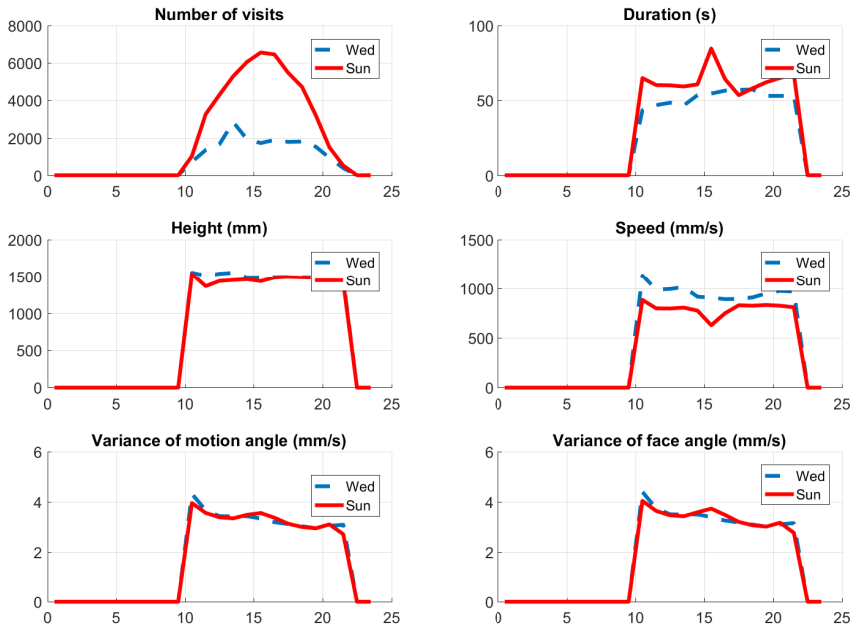


FIGURE 7-3 – Les métrique clés en Mercredi (bleu) et Dimanche(roux).

Figure 7-3 présente les métriques clés d'un mercredi et un dimanche. La ligne rouge est dimanche et la ligne bleue indique mercredi. Dans cette figure, on peut voir un nombre élevé de visites le dimanche par rapport au mercredi. Ceci est le résultat d'un grand nombre de visiteurs le week-end, alors que pendant la semaine, l'endroit est principalement occupé par la personne qui travaille dans le bâtiment. On peut également reconnaître une moyenne élevée de la vitesse de déplacement du mercredi au dimanche. Cela pourrait s'expliquer par la vitesse de déplacement des employés qui sont plus rapides que les visiteurs.

Ensuite, après l'analyse, on recrée l'ensemble classificateur pour cet ensemble de données. La figure 7-4 présente le résultat de la formation et de la détection des ensemble classificateurs en utilisant des métriques clés. Ce modèle utilisait des données provenant de

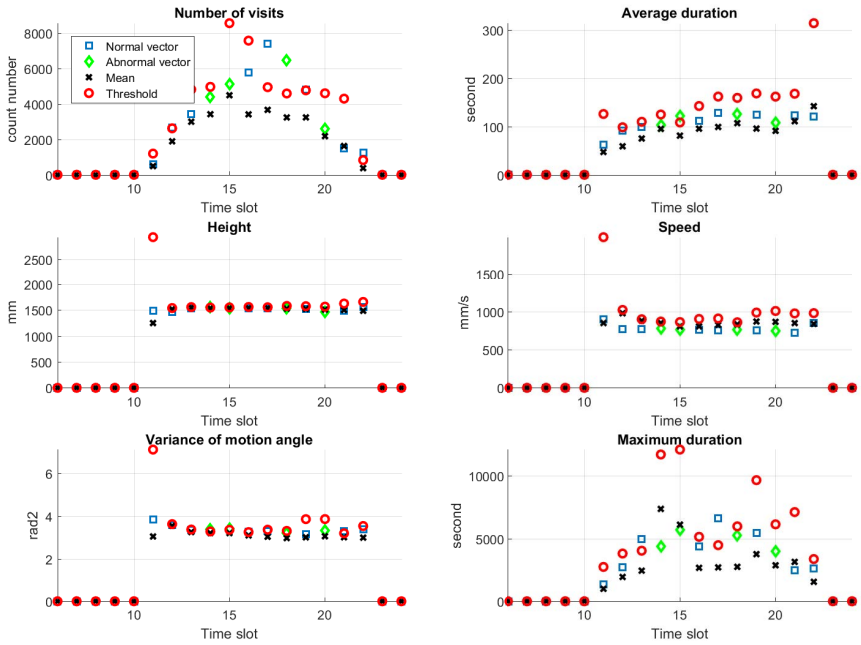


FIGURE 7-4 – La détection de l'ensemble classificateur pour le cas d'usage ATC.

3 dimanches de novembre et de décembre 2012. Le jour du test est le dimanche qui a des trajectoires anormales dans l'ensemble de données.

7.5 La détection des trajectoires atypiques

7.5.1 L'introduction

Dans cette section, on discute de la méthode DTA proposée pour le projet VIRTUALIS. La motivation de cette méthode vient de l'idée de créer un modèle de détection d'anomalie non supervisé à partir des trajectoires normales et de détecter les trajectoires très différentes de l'ensemble d'apprentissage. Comme la section 7.4, notre méthode sera appliquée à deux ensembles de données correspondant à deux projets VIRTUALIS et ATC. La méthodologie commune est présentée en premier. Ensuite, on présente la mise en œuvre de chaque scénario VIRTUALIS et ATC respectivement. Dans chaque scénario, l'extraction de la séquence, puis la fonction de dissimilarité correspondante seront représenté. Trois

noyaux GDTW, DTA, GA qui sont spécialisés pour le modèle seul-classe SVM et les données de séquence sont appliquées pour comparer leurs performances. Dans les expériences, on a prouvé que notre modèle peut détecter des séquences anormales avec un faible taux de fausses alarmes.

7.5.2 La méthodologie

Avec cette méthodologie, le détecteur doit être formé et peut être utilisé pour déduire la détection sur de nouvelles données. Premièrement, on définisse un ensemble de données d'apprentissage pour former le détecteur seul-classe SVM. L'ensemble de données contient des séquences extraites des observations de la caméra. Le processus de l'apprentissage est hors ligne. Une fois le détecteur formé, il est validé sur un jeu de données étiqueté. Ce processus vise à choisir les hyperparamètres optimaux pour le détecteur. Le processus de détection pourrait être fait en mode en ligne et hors ligne. La détection en ligne analyse les séquences en temps réel et les processus de détection hors ligne des séquences entièrement capturées. Ce processus déploie le détecteur entraîné dans le système de sécurité pour détecter des séquences anormales.

Dans le processus de l'apprentissage, on extrait des séquences d'activités à partir de trajectoires brutes de piétons dans des jours typiques en séparant d'abord le plan du bâtiment en plusieurs sous-zones. Ensuite, pour chaque trajectoire, les points de mouvement dans chaque sous-zone sont utilisés pour calculer les caractéristiques des activités. Et enfin les activités d'une trajectoire sont connectées par ordre chronologique pour créer une séquence d'activités. Ainsi, toute la trajectoire est cartographiée dans une séquence d'activité à travers ce processus d'extraction de séquence. Ensuite, on coupe ces séquences en parties plus petites par une fenêtre temporelle avec une durée maximale. Une mesure de similarité de séquence personnalisée est proposée en tant que noyau. En utilisant cette similarité, on peut calculer la matrice de Gram du noyau pour fournir le modèle seul-classe SVM.

Dans l'étape de validation, un ensemble de séquences étiquetées est utilisé pour tester le détecteur entraîné. Les étiquettes prédites de ces séquences sont données par le modèle seul-classe SVM pour identifier la séquence anormale. Les hyperparamètres sont calibrés pour avoir le résultat de classification optimal sur cet ensemble. La mesure d'évaluation est

le bon taux de détection et le taux de fausses alarmes.

Dans le processus de détection, on transforme les trajectoires enregistrées en temps réel en nouvelles séquences d'activités. Ensuite, la matrice de similarité entre les séquences dans l'ensemble d'apprentissage et l'ensemble de nouvelles séquences est calculée en utilisant la fonction de similarité personnalisée et les paramètres qui ont été utilisés pour l'apprentissage. Cette matrice est ensuite transformée en matrice Gram pour la prédiction. Comme le nombre de séquences en temps réel n'est pas important, ce processus peut être effectué après chaque période de 5 minutes.

Dans ce rapport, les expériences exécutent le processus de l'apprentissage et d'évaluation. L'étape de détection a été mise en œuvre lors de la démonstration au laboratoire ThereSIS sur le site de Thales, et il n'y a pas d'étape de détection pour le cas ATC.

7.5.3 Le cas d'usage VIRTUALIS

La description

Dans cette section, on présente une approche d'apprentissage automatique pour analyser automatiquement les trajectoires de personnes à l'intérieur d'un centre commercial. La méthodologie est présentée dans la dernière section. Cette section décrit davantage le processus de construction de séquence et la fonction de similarité personnalisée. L'expérience de cette approche est basée sur des données simulées parce qu'on ne peut pas construire la séquence des activités avec le donnée réel.

Le plan division

La représentation graphique de la construction simulée par Thales est présentée dans la figure 7-5. Chaque nœud de ce graphique représente une zone spécifique, où les gens peuvent faire des activités. Les arêtes relient les zones entre elles, ce qui montre les possibilités de passer d'un noeud à ses voisins. Les nœuds verts de la figure 7-5 affichent les sous-zones de surveillance, et les nœuds blancs sont des zones non surveillées. Les points rouges sont des éléments critiques tels que le serveur ou les imprimantes.

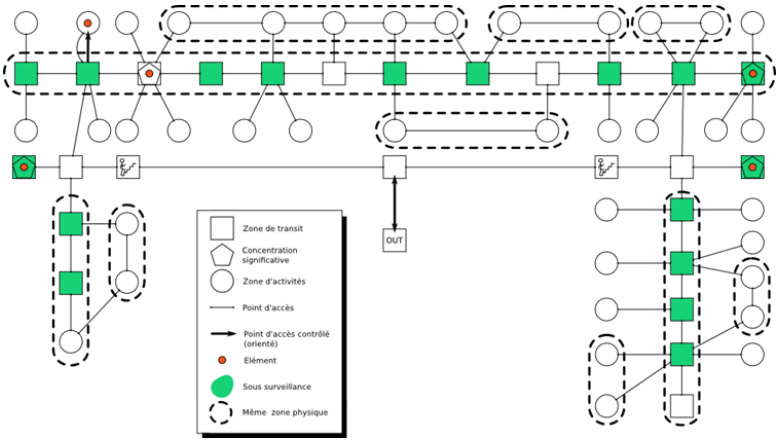


FIGURE 7-5 – La division des zones dans le bâtiment de Thalès.

Les séquences des activités

Pour construire la séquence des activités, il faut créer la définition d'une activité.

Definition 7.5.1. *L'activité est une action d'une personne dans un lieu spécifique. Cet endroit peut être sous surveillance ou non. Le type d'action est déterminé par le type de la sous-zone. Une activité est créée lorsqu'une personne reste au même endroit pendant plus longtemps un seuil prédéfini. Chaque activité est définie comme un vecteur* $A \in R^{(1\times4)}$, $A = [A^{(1)}, A^{(2)}, A^{(3)}, A^{(4)}]$ *où les quatre caractéristiques sont le ID de la personne, le type de lieu, le instant du début de l'activité et son durée respectivement.*

Après l'extraction des activités, on combine les activités de chaque personne en séquences d'activités et les représentons comme des séquences. Les séquences sont coupées par un fenêtre glissant du temps pour limiter la longueur d'une séquence - typiquement 2 ou 3 heures maximum. La séquence d'activité est représentée par S. $S \in R^{M\times4}$: $S = [A_1, A_2, ..., A_i, ..., A_M]$, $i \in [1, M]$. A_i signifie l'activité i^{th} de la séquence S où M est le nombre d'activités dans la séquence S.

La fonction de similarité et le noyau

Après la création des séquences des activités, on construit une matrice de noyau pour faire l'apprentissage des la modèle seul-classe SVM. Dans la premier étape, on définie une fonction de similarité personnalisée pour calculer la distance entre deux événements. Cette fonction est composée de 4 parties : le coût d'identification, le coût du type d'emplacement, le coût en temps et le coût en durée. Le coût total est alors trouvé par une équation linéaire de tous ces coûts. La fonction de dissemblance personnalisée $DissAct$ est présentée :

$$DissAct(A_i, A_j) = \sum_{k=1}^{4} \theta_k C_k(A_i^{(k)}, A_j^{(k)}) \tag{7.1}$$

Avec cette fonction, trois noyaux GDTW, DTA et GA kernel sont appliqués pour calculer la noyau matrice entre des séquences. Cette matrice de noyau est utilisée pour la modèle seul-classe SVM.

L'expérimentes

Pour évaluer les performances de notre modèle, on utilise un ensemble de test de 300 séquences normales et de 119 séquences anormales. Les séquences anormales proviennent de trois scénarios d'attaque. La modèle seul classe SVM est entrainé par un ensemble de 500 séquences étiquetés. La tableau 7.4 représente le meilleur performance de chaque noyau. Selon ce tableau, le noyau GA a la meilleure précision, et le faux positif est maintenu très bas (5 %).

TABLE 7.4 – Performance of different kernels of seul classe SVM classifier.

Kernel	σ	μ	Acc	TPR	FPR
GDTW	200	0.1	0.84	0.6050	0.06
DTAK	0.1	0.1	0.8807	0.6555	0.03
GA	0.01	0.1	0.926	0.8739	0.0533

Il y a trois trajectoires dans l'ensemble des séquences anormal. Deux d'entre eux sont plus faciles à détecter que celui qui correspond à une étape de préparation de l'attaquant, donc 60% du taux de détection est acceptable pour cette expérience, et tous les noyaux

semblent le satisfaire.

7.5.4 Le cas d'usage ATC

La deuxième partie de ce chapitre présente l'implémentation de la technique DTA pour le scénario du centre commercial dans le projet ATC. Dans ce projet, le problème de ré-identification est mieux résolu en utilisant un système de caméra 3D, et la plupart de l'endroit dans le bâtiment est sous la surveillance de la caméra. Ainsi, la trajectoire de chaque personne est entièrement suivie à l'intérieur du bâtiment. Cependant, on ne peut pas mettre toute la méthode DTA définie dans le scénario de bureau ci-dessus pour résoudre ce problème car il n'y a pas de sous-zones de division naturelles. Alors c'est impossible d'extraire des activités en utilisant le plan naturel du bâtiment.

Par conséquent, la méthode DTA est modifiée pour ce scénario. La première modification est dans la création du graphique, puis la définition de la fonction d'activité et de dissimilarité est modifiée en raison de la structure différente de la trajectoire dans l'ensemble de données ATC.

La division des zones

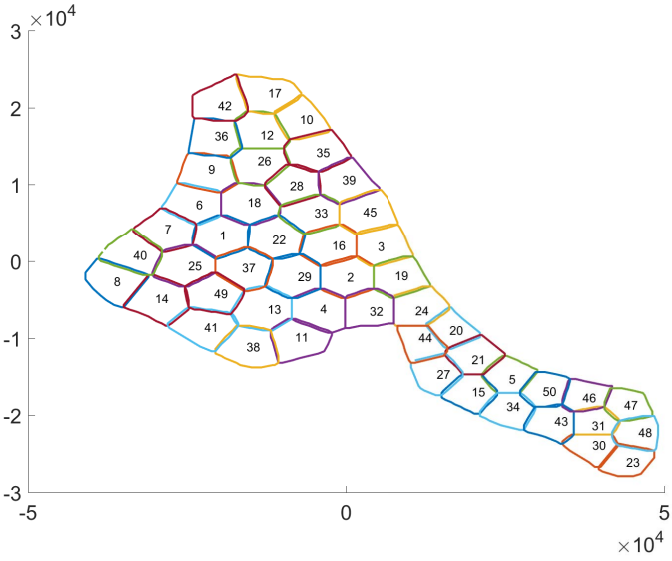


FIGURE 7-6 – La division par K-means clustering.

Pour créer des sous-zones, on utilise le méthode k-Means pour diviser le bâtiment à 50 sous-zones de taille égale comme la figure 7-6. Ces sous-zones sont utilisés pour créer les activités dans l'étape suivant.

Les séquences des activités

Après avoir séparé les zones, on extrait la trajectoire à la séquence de la zone où chaque élément de cette séquence est un vecteur de 7 caractéristiques : $A \in R^{1\times 7}$. Où $A^{(1)}$ est l'identité de la personne, $A^{(2)}$ est l'heure à laquelle il entre dans une zone, $A^{(3)}$ est sa durée à l'intérieur de la zone, $A^{(4)}$ est la vitesse moyenne dans cette zone, $A^{(5)}$ est l'identité de la zone. $A^{(6)}$ indique la variation de son mouvement d'angle, et $A^{(7)}$ est le rapport de sa distance réellement déplacée sur son passage réel.

En raison de la modification de la structure des séquences, la fonction de similarité est également modifiée pour s'adapter à la nouvelle structure de la séquence. La fonction $DissAct$ dans l'équation 7.1 est modifie de 4 à 7 éléments. Les trois noyaux sont rappliqué pour comparaison dans ce cas.

L'expérimentes

Les expériences de cette section pourraient être séparées en deux parties différentes. Dans la première expérience, on forme un classificateur supervisé sur le même jeu de données étiqueté avec le modèle de l'auteur dans [18]. Ensuite, on compare les performances de ce classificateur avec le modèle proposé par l'auteur. Cette expérience vise à mesurer les performances de la méthode ATD en utilisant le même ensemble de données que l'auteur pour la tâche de détection d'anomalie.

La deuxième expérience réplique l'expérience dans le scénario VIRTUALIS sur l'ensemble de données ATC qui forme un classificateur non supervisé sur l'ensemble de données non étiqueté et tester le modèle sur l'ensemble de données étiqueté pour comparer les performances des différents noyaux. Cette expérience étudie la capacité de la méthode proposée pour le cas du centre commercial.

Les expériences de cette section sont effectuées en utilisant deux ensembles de trajectoires : ensemble de données étiqueté et non étiqueté. Le premier est un ensemble de trajectoires étiquetées qui contient 29 trajectoires anormales et 29 trajectoires normales étiquetées par un expert. L'anomalie d'une trajectoire provient d'une personne qui se perd et peut avoir besoin d'aide. Le deuxième ensemble de données est un ensemble de trajectoires non étiquetées. Cet ensemble de données est un sous-ensemble de l'énorme ensemble de données ATC, il contient 300 trajectoires aléatoires.

TABLE 7.5 – La performance des noyaux différents de la classificateur seul-classe SVM.

Kernel	σ	C	Accuracy	TPR	FPR
GDTW	300	100	0.9138	1	0.1724
GDTW	300	10000	0.9138	1	0.1724
DTA	10	10	0.8103	0.7586	0.1379
DTA	1	100	0.8103	0.8276	0.1724
GA	10	0.1	0.8448	0.8276	0.1379
GA	10	100	0.8448	0.8276	0.1379
Brvsvcic [18]			0.9138	0.931	0.103

La table 7.5 présente les performances des noyaux à différents couples d'hyperparamètres σ et C qui donnent la précision optimale de chaque noyau. Dans cette table, le noyau GDTW donne la meilleure précision, et comparable au résultat de l'auteur dans [18] où la précision est d'environ 90 %. De plus, on a utilisé le test LOO, donc notre modèle n'est pas "overfit" à l'ensemble de données. Les autres noyaux (GA et DTA) ont des résultats très importants, et le compromis entre le vrai taux de détection (TPR) et le taux de fausses alarmes (FPR) est montré pour une valeur différente de σ et C.

La deuxième expérimente tend à créer un classificateur non supervisé qui utilise le jeu de données non étiqueté pour former le classificateur one-class SVM et valider le modèle par l'ensemble de données étiqueté.

La figure 7-7 présente la comparaison des performances entre trois noyaux. Dans cette figure, on a sélectionné le plus faible taux FPR possible lorsque le vrai taux de détection est d'au moins 60%. En fait, on cherche le moins de fausse alarme possible parmi les noyaux pour avoir un taux de détection acceptable. Selon la figure, la moindre fausse alarme se produit avec le noyau GDTW à $\sigma = 10$ où le taux de fausses alarmes est d'environ 5%. Le noyau GA ne fonctionne pas bien comme prévu.

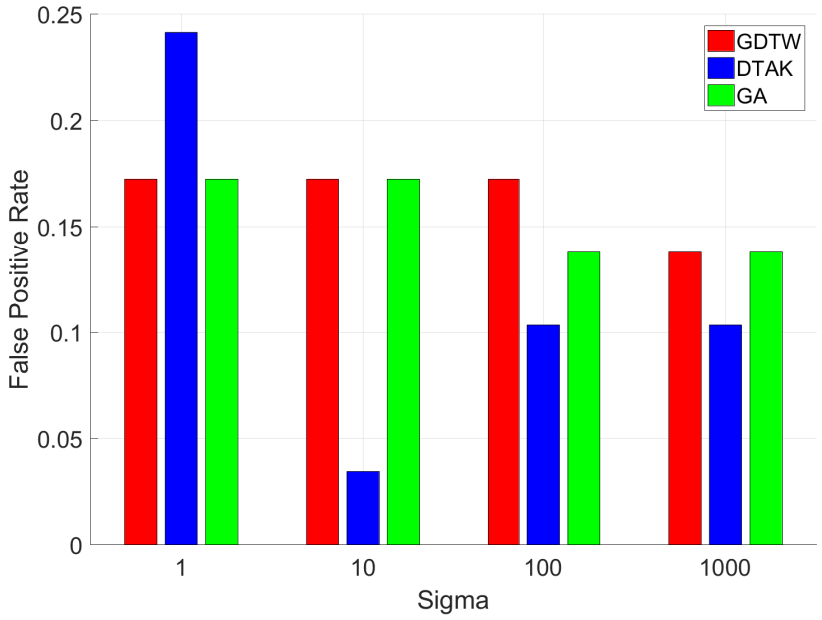


FIGURE 7-7 – La comparaison des noyaux.

7.6 La conclusion et la perspectives

7.6.1 La conclusion

Cette thèse est issue de la collaboration entre l'équipe M2S du laboratoire ICD et les autres membres du projet VIRTUALIS. L'objectif principal de la thèse est de participer à l'analyse des risques SP3-Global, un sous-projet du projet VIRTUALIS, pour développer des méthodes de détection d'anomalies à partir de données provenant de capteurs et de caméras dans un bâtiment. Ces méthodes pourraient analyser ces données pour détecter le suspect (terroriste, attaquants) automatiquement.

L'ensemble de données VIRTUALIS provient d'un modèle de simulation et d'un système de caméras et autres capteurs mis en place sur le site de Thales. Au départ, le consortium avait tendance à appliquer le système de sécurité pour deux scénarios : le bâtiment commercial sur le site de Thales et la station de métro Chatelet. Cependant, en raison de l'indisponibilité de l'ensemble de données de la station de métro, on remplace ce scénario par l'ensemble de données ATC. L'ensemble de données ATC est un ensemble de données

publiques obtenu par un système de caméras 3D au centre commercial ATC d'Osaka, au Japon. Le but de cet ensemble de données est différent de celui de VIRTUALIS car il tend à aider les personnes qui se perdent et ont besoin d'aide dans le bâtiment. Une limitation existe dans le jeu de données réel VIRTUALIS, où la tâche de ré-identification n'a pas été faite parfaitement. Ainsi, les méthodes qui fonctionnent avec la séquence ne peuvent pas utiliser cet ensemble de données. En d'autres termes, cette thèse a exploré les données de trajectoire des piétons dans les lieux publics. Il existe deux scénarios : le bâtiment commercial dans le jeu de données VIRTUALIS et le centre commercial dans le jeu de données ATC. Dans l'ensemble de données VIRTUALIS, il existe un jeu de données simulé et un jeu de données réel, un scénario d'attaque a été préparé par des experts en sécurité pour tester nos modèles. Ainsi, seul l'ensemble de données réel est disponible dans l'ensemble de données ATC. Avec ces jeux de données, on a développé deux méthodes pour le projet VIRTUALIS. La première méthode DEA analyse les trajectoires en tant qu'événements dans une zone du bâtiment. La seconde méthode DTA extrait les trajectoires en séquences d'activités et construit un détecteur pour ce type de séquence.

La technique DEA est basée sur la méthode paramétrique pour le problème d'anomalie de détection. Cette technique sélectionne les zones critiques dans le bâtiment et divise un jour en plusieurs fenêtres temporelles. Ensuite, les trajectoires dans ces sous-zones dans chaque fenêtre temporelle sont transformées en métriques clés. Ensuite, un ensemble classificateur qui calcule un seuil pour chaque mesure clé dans chaque fenêtre temporelle est créé, et ces seuils sont utilisés pour détecter des métriques clés anormales. Une autre approche des métriques clés consiste à construire un seul classificateur en utilisant le vecteur des métriques clés dans chaque fenêtre temporelle en entrée. Le classificateur seul-classe SVM est ensuite implémenté pour détecter un vecteur anormal de métriques clés. L'expérimentation montre un résultat intéressant pour le scénario VIRTUALIS où le ensemble classificateur peut détecter les événements d'attaque dans les ensembles de données simulés et réels et le classificateur unique donne des résultats de détection considérables. Dans l'ensemble de données ATC, parce que l'ensemble de données étiqueté est sélectionné par l'expérience d'un expert, et qu'il existe de nombreuses trajectoires anormales dans l'ensemble de données non étiqueté, cette méthode ne fonctionne pas bien.

La technique DTA transforme les séries temporelles des points de mouvement dans

les trajectoires en séquences d'activités. Dans l'ensemble de données VIRTUALIS, ils sont simplement déduits de l'emplacement. Les activités sont créées efficacement en utilisant les séparations de bureaux dans le bâtiment. Au contraire, l'ensemble de données ATC étant un espace ouvert, l'étape d'extraction est réalisée en divisant la zone entière en sous-zones de taille égale et la séquence d'activités est considérée comme la séquence de sous-zones. Après avoir extrait les séquences, on utilise la classificateur non supervisé One Class SVM. Une fonction de dissimilarité spécifique entre les séquences et les noyaux est définie pour appliquer des séquences pour ce classificateur. Trois types de noyau tels que GDTW, DTA et GA sont appliqués. L'algorithme d'alignement du noyau est utilisé pour rechercher les hyperparamètres optimaux. Les expériences du scénario VIRTUALIS montrent une grande performance pour l'ensemble de données simulé. Dans l'ensemble de données ATC, on a fait un test pour comparer notre modèle avec l'auteur de cet ensemble de données et on a obtenu un bon résultat. Ensuite, on recrée le test du modèle one-class SVM non supervisé pour comparer les performances des différents noyaux.

Ces deux techniques ont leur avantage et leur inconvénient. Pour un système de sécurité, ils couvrent la faiblesse de l'autre. La technique DEA est simple et fonctionne dans une zone critique, donc seule la sous-trajectoire à l'intérieur de la zone est supervisée, et la technique DTA analyse l'ensemble de la trajectoire. Cependant, la technique DTA est limitée par le problème de ré-identification, ce qui signifie que toute la trajectoire d'une personne est interrompue et que on ne peut pas avoir toute la trajectoire dans un jeu de données réel. Le DEA n'est pas affecté par ce problème.

7.6.2 La perspective

À l'avenir, même si le projet VIRTUALIS est terminé, de nombreux problèmes doivent encore être résolus. Le premier problème est l'amélioration de la détection des piétons, la reconnaissance et la ré-identification du piéton avec une grande précision dans un lieu public. Ce problème est difficile en raison de nombreuses raisons telles que l'environnement instable (condition de lumière), trop de personnes dans le champ de vision de la caméra (scénario de station de métro). Le deuxième problème est l'échelle du modèle lorsque la taille de l'ensemble de données est plus grande. Le seul-classe SVM est assez mauvais

avec le problème d'échelle, il prend un long temps de calcul pour la matrice gramme en raison de la fonction de dissimilarité personnalisée. Par exemple, s'il y a 1000 séquences dans l'ensemble d'apprentissage, chaque séquence a une moyenne de 5 activités, donc la fonction de coût est appelée $1000^2 * 5^2 = 25 * 10^6$ fois. Ainsi, la quantité croissante de séquence d'entraînement causera plus de problèmes pour entraîner le modèle.

Pour la méthode DEA, la technique pourrait être améliorée en spécifiant le modèle pour chaque catégorie, telle que la fonctionnalité humaine, période. Par exemple, si on a plus de jours dans l'ensemble de données VIRTUALIS, on peut calculer des seuils spécifiques pour chaque mesure clé pour chaque période, par exemple pour chaque jour d'une semaine ou période spéciale d'une année. Si on peut reconnaître l'identité de chaque personne à travers la caméra, on peut comparer leur rôle (ingénieur, secrétaire, invité) dans une liste de référence, et appliquer le modèle spécifique pour eux (classificateur pour ingénieur, secrétaire).

Il y a aussi beaucoup de place pour améliorer la méthode DTA. Tout d'abord, à partir de l'étape de pré-traitement, il existe d'autres méthodes pour représenter une trajectoire telle qu'un tenseur ou une matrice comme introduit dans [131]. Avec cette représentation, on peut calculer la similarité entre chaque personne dans le bâtiment en utilisant des méthodes telles que le filtrage collaboratif, et donc identifier les anomalies. La deuxième amélioration pourrait être faite pour changer la mesure de similarité. Jusqu'à présent, il s'agit d'une fonction avec un choix arbitraire d'hyperparamètres. Si on peut définir une meilleure fonction qui peut comparer la distance entre deux activités, le résultat de la détection sera amélioré.

Hors de la portée des méthodes proposées, en raison d'une grande tendance à utiliser Deep Neural Network pour résoudre les problèmes d'apprentissage machine, on devrait essayer d'appliquer le réseau neuronal récurrent (RNN), en particulier la mémoire à court terme (LSTM) pour résoudre notre problème de séquence anormale . En fait, comme notre ensemble de données est principalement une trajectoire, on peut former un modèle pour prédire le mouvement suivant de chaque peuple, et si sa trajectoire est très différente de notre prédiction, elle est classée comme une anomalie.

Bibliographie

[1] Agarwal, D. (2005). An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE.

[2] Agarwal, D. (2007). Detecting anomalies in cross-classified streams : a bayesian approach. *Knowledge and information systems*, 11(1) :29–44.

[3] Agrawal, S. and Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60 :708–713.

[4] Ahmed, R., Temko, A., Marnane, W. P., Boylan, G., and Lightbody, G. (2017). Exploring temporal information in neonatal seizures using a dynamic time warping based svm kernel. *Computers in biology and medicine*, 82 :100–110.

[5] Ahmed, T., Pedersen, T. B., Calders, T., and Lu, H. (2016). Online risk prediction for indoor moving objects. In *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, volume 1, pages 102–111. IEEE.

[6] Antonakaki, P., Kosmopoulos, D., and Perantonis, S. J. (2009). Detecting abnormal human behaviour using multiple cameras. *Signal Processing*, 89(9) :1723–1738.

[7] Araujo, R. and Kamel, M. S. (2014). Semi-supervised kernel-based temporal clustering. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 123–128. IEEE.

[8] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3) :337–404.

[9] Aven, T. (2015). *Risk analysis*. John Wiley & Sons.

[10] Bahlmann, C., Haasdonk, B., and Burkhardt, H. (2002). Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*, pages 49–54. IEEE.

[11] Baseer, S. (2013). Heterogenous networks architectures and their security weaknesses. *International Journal of Computer and Communication Engineering*, 2(2) :90.

[12] Bedagkar-Gala, A. and Shah, S. K. (2014). A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4) :270–286.

[13] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

[14] Bhinge, S., Levin-Schwartz, Y., and Adalı, T. (2017). Data-driven fusion of multicamera video sequences : Application to abandoned object detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 1697–1701. IEEE.

[15] Boissier, O. and Tessier, C. (2013). Dossier éthique et ia. *Association Française pour l'Intelligence Artificielle*, page 4.

[16] Bonhomme, S., Campo, E., Esteve, D., and Guennec, J. (2007). An extended prosafe platform for elderly monitoring at home. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4056–4059. IEEE.

[17] Boukhechba, M., Bouzouane, A., Bouchard, B., Gouin-Vallerand, C., and Giroux, S. (2015). Online prediction of people's next point-of-interest : Concept drift support. In *Human Behavior Understanding*, pages 97–116. Springer.

[18] Brščić, D., Ikeda, T., and Kanda, T. (2017). Do you need help ? a robot providing information to people who behave atypically. *IEEE Transactions on Robotics*, 33(2) :500–506.

[19] Brscic, D., Kanda, T., Ikeda, T., and Miyashita, T. (2013). Person tracking in large public spaces using 3-d range sensors. *IEEE Transactions on Human-Machine Systems*, 43(6) :522–534.

[20] Buczak, A. L. and Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2) :1153–1176.

[21] Cai, Y., Wang, H., Chen, X., and Jiang, H. (2015). Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. *IET intelligent transport systems*, 9(8) :810–816.

[22] Camargo, J. E. and González, F. A. (2009). A multi-class kernel alignment method for image collection summarization. In *Iberoamerican Congress on Pattern Recognition*, pages 545–552. Springer.

[23] Camps-Valls, G., Tuia, D., Bruzzone, L., and Benediktsson, J. A. (2014). Advances in hyperspectral image classification : Earth monitoring with statistical learning methods. *IEEE signal processing magazine*, 31(1) :45–54.

[24] Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2) :1.

[25] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection : A survey. *ACM computing surveys (CSUR)*, 41(3) :15.

[26] Chen, Y. and Wu, W. (2017). Application of one-class support vector machine to quickly identify multivariate anomalies from geochemical exploration data. *Geochemistry : Exploration, Environment, Analysis*, 17(3) :231–238.

[27] Chen, Y.-C., Zhu, X., Zheng, W.-S., and Lai, J.-H. (2018). Person re-identification by camera correlation aware feature augmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(2) :392–408.

[28] Chen, Z., Shen, H. T., and Zhou, X. (2011). Discovering popular routes from trajectories. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 900–911. IEEE.

[29] Cohen, G., Hilario, M., and Pellegrini, C. (2004). One-class support vector machines with a conformal kernel. a case study in handling class imbalance. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 850–858. Springer.

[30] Cohen, M. D. and Axelrod, R. (2000). *Harnessing complexity : organizational implications of a scientific frontier*. Simon and Schuster.

[31] Consortium, V. et al. D2.3 prototype de réseau de capteurs virtualisés. *VIRTUALIS project, FUI*, 15.

[32] Consortium, V. et al. Fiche technique de projet systematic paris-region groupe thématique cns. *VIRTUALIS project, FUI*, 15.

[33] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3) :273–297.

[34] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. (2002). On kernel-target alignment. In *Advances in neural information processing systems*, pages 367–373.

[35] Cuturi, M. (2011). Fast global alignment kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 929–936.

[36] Cuturi, M., Vert, J.-P., Birkenes, O., and Matsui, T. (2007). A kernel for time series based on global alignments. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–413. IEEE.

[37] Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*, (2) :222–232.

[38] Deshpande, N. T. and Ravishankar, S. (2017). Face detection and recognition using viola-jones algorithm and fusion of pca and ann. *Advances in Computational Sciences and Technology*, 10(5) :1173–1189.

[39] Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11).

[40] Dorveaux, E. (2011). *Magneto-inertial navigation : principles and application to an indoor pedometer*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[41] Emadi, H. S. and Mazinani, S. M. (2018). A novel anomaly detection algorithm using dbscan and svm in wireless sensor networks. *Wireless Personal Communications*, 98(2) :2025–2035.

[42] Erfani, S. M., Rajasegarar, S., Karunasekera, S., and Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58 :121–134.

[43] Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer.

[44] Falliere, N., Murchu, L. O., and Chien, E. (2011). W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5(6) :29.

[45] Fennelly, L. (2016). *Effective physical security*. Butterworth-Heinemann.

[46] Fisher, W. D., Camp, T. K., and Krzhizhanovskaya, V. V. (2017). Anomaly detection in earth dam and levee passive seismic data using support vector machines and automatic feature selection. *Journal of Computational Science*, 20 :143–153.

[47] Fod, A., Howard, A., and Mataric, M. (2002). A laser-based people tracker. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 3024–3029. IEEE.

[48] Foundations, C. (1996). Protecting america's infrastructures, the report of the president's commission on critical infrastructure protection, october 1997. *National Defense University, Washington DC, August*.

[49] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

[50] Fu, C., Zhang, P., Jiang, J., Yang, K., and Lv, Z. (2017). A bayesian approach for sleep and wake classification based on dynamic time warping method. *Multimedia Tools and Applications*, 76(17) :17765–17784.

[51] Glas, D. F., Miyashita, T., Ishiguro, H., and Hagita, N. (2009). Laser-based tracking of human position and orientation using parametric shape modeling. *Advanced robotics*, 23(4) :405–428.

[52] Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23) :e215–e220.

[53] Graepel, T. and Obermayer, K. (1998). Fuzzy topographic kernel clustering. In *Proceedings of the 5th GI Workshop Fuzzy Neuro Systems*, volume 98, pages 90–97.

[54] Gudmundsson, S., Runarsson, T. P., and Sigurdsson, S. (2008). Support vector machines and dynamic time warping for time series. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2772–2776. IEEE.

[55] Hasan, M. A. M., Nasser, M., Pal, B., and Ahmad, S. (2014). Support vector machine and random forest modeling for intrusion detection system (ids). *Journal of Intelligent Learning Systems and Applications*, 6(01) :45.

[56] Hoque, E., Dickerson, R. F., Preum, S. M., Hanson, M., Barth, A., and Stankovic, J. A. (2015). Holmes : A comprehensive anomaly detection system for daily in-home activities. In *Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on*, pages 40–51. IEEE.

[57] Hsu, Y.-L., Chu, C.-L., Tsai, Y.-J., and Wang, J.-S. (2015). An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition. *IEEE Sensors Journal*, 15(1) :154–163.

[58] Igel, C., Glasmachers, T., Mersch, B., Pfeifer, N., and Meinicke, P. (2007). Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2).

[59] Izakian, H., Pedrycz, W., and Jamal, I. (2015). Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39 :235–244.

[60] Jain, A., Nandakumar, K., and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12) :2270–2285.

[61] Jain, A. K. and Gupta, B. (2017). Phishing detection : analysis of visual similarity based approaches. *Security and Communication Networks*, 2017.

[62] Jansson, D., Rosén, O., and Medvedev, A. (2015). Parametric and nonparametric analysis of eye-tracking data by anomaly detection. *IEEE Transactions on Control Systems Technology*, 23(4) :1578–1586.

[63] Jeni, L. A., Lőrincz, A., Szabó, Z., Cohn, J. F., and Kanade, T. (2014). Spatio-temporal event classification using time-series kernel based structured sparsity. In *European Conference on Computer Vision*, pages 135–150. Springer.

[64] Joder, C., Essid, S., and Richard, G. (2008). Alignment kernels for audio classification with application to music instrument recognition. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE.

[65] Kanagala, H. K. and Krishnaiah, V. J. R. (2016). A comparative study of k-means, dbscan and optics. In *Computer Communication and Informatics (ICCCI), 2016 International Conference on*, pages 1–6. IEEE.

[66] Kfir, Z. and Wool, A. (2005). Picking virtual pockets using relay attacks on contactless smartcard. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 47–58. IEEE.

[67] Kulkarni, A., Pino, Y., French, M., and Mohsenin, T. (2016). Real-time anomaly detection framework for many-core router through machine-learning techniques. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(1) :10.

[68] Laxhammar, R. and Falkman, G. (2014). Online learning and sequential anomaly detection in trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 36(6) :1158–1173.

[69] Le, V.-K., Grall-Maes, E., and Beauseroy, P. (2018). Abnormal events detection for infrastructure security using key metrics. In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods - Volume 1 : ICPRAM,*, pages 284–290. INSTICC, SciTePress.

[70] LE, V.-K., Pierre, B., and Grall-Maes, E. (2018). Abnormal trajectory detection for security infrastructure. In *Proceedings of the 2nd International Conference on Digital Signal Processing*, pages 1–5. ACM.

[71] Li, M., Liu, X., Wang, L., Dou, Y., Yin, J., and Zhu, E. (2016). Multiple kernel clustering with local kernel alignment maximization.

[72] Lin, H., Deng, J. D., and Woodford, B. J. (2015a). Anomaly detection in crowd scenes via online adaptive one-class support vector machines. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2434–2438. IEEE.

[73] Lin, W.-C., Ke, S.-W., and Tsai, C.-F. (2015b). Cann : An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, 78 :13–21.

[74] Liu, W., Zheng, Y., Chawla, S., Yuan, J., and Xing, X. (2011). Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1010–1018. ACM.

[75] Lu, Y., Wang, L., Lu, J., Yang, J., and Shen, C. (2014). Multiple kernel clustering based on centered kernel alignment. *Pattern Recognition*, 47(11) :3656–3664.

[76] Melo, J., Naftel, A., Bernardino, A., and Santos-Victor, J. (2006). Detection and classification of highway lanes using vehicle motion trajectories. *IEEE Transactions on intelligent transportation systems*, 7(2) :188–200.

[77] Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3) :90–126.

[78] Morris, B. T. and Trivedi, M. M. (2008a). Learning and classification of trajectories in dynamic scenes : A general framework for live video analysis. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 154–161. IEEE.

[79] Morris, B. T. and Trivedi, M. M. (2008b). A survey of vision-based trajectory learning and analysis for surveillance. *IEEE transactions on circuits and systems for video technology*, 18(8) :1114–1127.

[80] Münz, G., Li, S., and Carle, G. (2007). Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet*.

[81] Negri, P. and Garayalde, D. (2017). Pedestrian tracking using probability fields and a movement feature space. *Dyna*, 84(200) :217–227.

[82] Nguyen, K., Fookes, C., Ross, A., and Sridharan, S. (2017). Iris recognition with off-the-shelf cnn features : A deep learning perspective. *IEEE Access*.

[83] Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 344–353. ACM.

[84] Pang, L. X., Chawla, S., Liu, W., and Zheng, Y. (2011). On mining anomalous patterns in road traffic streams. In *International Conference on Advanced Data Mining and Applications*, pages 237–251. Springer.

[85] Pang, L. X., Chawla, S., Liu, W., and Zheng, Y. (2013). On detection of emerging anomalous traffic patterns using gps data. *Data & Knowledge Engineering*, 87 :357–373.

[86] Parmar, J. D. and Patel, J. T. (2017). Anomaly detection in data mining : A review. *International Journal*, 7(4).

[87] Pelekis, N., Theodoridis, Y., and Janssens, D. (2014). On the management and analysis of our lifesteps. *ACM SIGKDD Explorations Newsletter*, 15(1) :23–32.

[88] Pena, M., Arratia, A., and Belanche, L. A. (2016). Multivariate dynamic kernels for financial time series forecasting. In *International Conference on Artificial Neural Networks*, pages 336–344. Springer.

[89] Petitjean, F., Forestier, G., Webb, G. I., Nicholson, A. E., Chen, Y., and Keogh, E. (2014). Dynamic time warping averaging of time series allows faster and more accurate classification. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 470–479. IEEE.

[90] Piérard, S. and Van Droogenbroeck, M. (2012). Estimation of human orientation based on silhouettes and machine learning principles. In *International conference on pattern recognition applications and methods (ICPRAM)*, pages 51–60.

[91] Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C. (2017). Video processing from electro-optical sensors for object detection and tracking in a maritime environment : A survey. *IEEE Transactions on Intelligent Transportation Systems*, 18(8) :1993–2016.

[92] Qin, A. K. and Suganthan, P. N. (2004). Kernel neural gas algorithms with application to cluster analysis. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 617–620. IEEE.

[93] Queiroz, L. P., Gomes, J. P. P., Rodrigues, F. C. M., Brito, F. T., Chaves, I. C., Leite, L. G., and Machado, J. C. (2018). Fault detection in hard disk drives based on a semi parametric model and statistical estimators. *New Generation Computing*, 36(1) :5–19.

[94] Ratsch, G., Mika, S., Scholkopf, B., and Muller, K.-R. (2002). Constructing boosting algorithms from svms : an application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9) :1184–1199.

[95] Ravale, U., Marathe, N., and Padiya, P. (2015). Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function. *Procedia Computer Science*, 45 :428–435.

[96] Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems*, 21(6) :11–25.

[97] Sadoughi, N. and Busso, C. (2015). Retrieving target gestures toward speech driven animation with meaningful behaviors. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 115–122. ACM.

[98] Saitoh, S. (1988). *Theory of reproducing kernels and its applications*, volume 189. Longman.

[99] Sakoe, H. and Chiba, S. (1970). A similarity evaluation of speech patterns by dynamic programming. In *Nat. Meeting of Institute of Electronic Communications Engineers of Japan*, page 136.

[100] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1) :43–49.

[101] Salem, O., Guerassimov, A., Mehaoua, A., Marcus, A., and Furht, B. (2016). Anomaly detection in medical wireless sensor networks using svm and linear regression models.

[102] Samorè, A., Guermandi, M., Placati, S., and Guerrieri, R. (2017). Parametric detection and classification of compact conductivity contrasts with electrical impedance tomography. *IEEE Transactions on Instrumentation and Measurement*, 66(10) :2666–2679.

[103] Schneider, N. and Gavrila, D. M. (2013). Pedestrian path prediction with recursive bayesian filters : A comparative study. In *German Conference on Pattern Recognition*, pages 174–183. Springer.

[104] Schöler, F., Behley, J., Steinhage, V., Schulz, D., and Cremers, A. B. (2011). Person tracking in three-dimensional laser range data with explicit occlusion adaption. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1297–1303. IEEE.

[105] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7) :1443–1471.

[106] Senin, P. (2008). Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855 :1–23.

[107] Shahid, N., Naqvi, I. H., and Qaisar, S. B. (2015). One-class support vector machines : analysis of outlier detection for wireless sensor networks in harsh environments. *Artificial Intelligence Review*, 43(4) :515–563.

[108] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

[109] Shawe-Taylor, J. and Žličar, B. (2015). Novelty detection with one-class support vector machines. In *Advances in Statistical Models for Data Analysis*, pages 231–257. Springer.

[110] Shen, M., Liu, D.-R., and Shann, S.-H. (2015). Outlier detection from vehicle trajectories to discover roaming events. *Information Sciences*, 294 :242–254.

[111] Shimodaira, H., Noma, K.-i., Nakai, M., and Sagayama, S. (2002). Dynamic time-alignment kernel in support vector machine. In *Advances in neural information processing systems*, pages 921–928.

[112] Shiravi, S., Hossain, K., Fu, L., and Ghods, A. (2016). Tra-963 : Evaluation of using wifi signals to estimate intersection travel time.

[113] Shrivastava, A., Patel, V. M., and Chellappa, R. (2014). Multiple kernel learning for sparse representation-based classification. *IEEE Transactions on Image Processing*, 23(7) :3013–3024.

[114] Snowden, E. (2014). Here's how we take back the internet. *TED Talk, March*.

[115] Soheily-Khah, S., Douzal-Chouakria, A., and Gaussier, E. (2016). Generalized k-means-based clustering for temporal data under weighted and kernel time warp. *Pattern Recognition Letters*, 75 :63–69.

[116] Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5) :631–645.

[117] Spinello, L., Luber, M., and Arras, K. O. (2011). Tracking people in 3d using a bottom-up top-down detector. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1304–1310. IEEE.

[118] Spinsanti, L., Berlingerio, M., and Pappalardo, L. (2013). Mobility and geo-social networks.

[119] Sprinthall, R. C. and Fisk, S. T. (1990). *Basic statistical analysis*. Prentice Hall Englewood Cliffs, NJ.

[120] Subudhi, S. and Panigrahi, S. (2015). Quarter-sphere support vector machine for fraud detection in mobile telecommunication networks. *Procedia Computer Science*, 48 :353–359.

[121] Taslimi-Renani, E., Modiri-Delshad, M., Elias, M. F. M., and Rahim, N. A. (2016). Development of an enhanced parametric model for wind turbine power curve. *Applied energy*, 177 :544–552.

[122] Tixier, J., Dusserre, G., Salvi, O., and Gaston, D. (2002). Review of 62 risk analysis methodologies of industrial plants. *Journal of Loss Prevention in the process industries*, 15(4) :291–303.

[123] Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., and Huerta, J. (2015). Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23) :9263–9278.

[124] Wang, T., Zhao, D., and Tian, S. (2015). An overview of kernel alignment and its applications. *Artificial Intelligence Review*, 43(2) :179–192.

[125] Wang, Y., Wu, K., and Ni, L. M. (2017). Wifall : Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2) :581–594.

[126] Weller-Fahy, D. J., Borghetti, B. J., and Sodemann, A. A. (2015). A survey of distance and similarity measures used within network intrusion anomaly detection. *IEEE Communications Surveys & Tutorials*, 17(1) :70–91.

[127] Yi, S., Li, H., and Wang, X. (2015). Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3488–3496.

[128] Zhang, Q., Wang, X., and Wang, X. (2015). An optics clustering-based anomalous data filtering algorithm for condition monitoring of power equipment. In *International Workshop on Data Analytics for Renewable Energy Integration*, pages 123–134. Springer.

[129] Zhang, Z., Huang, K., and Tan, T. (2006). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1135–1138. IEEE.

[130] Zhao, J. and Itti, L. (2016). Classifying time series using local descriptors with hybrid sampling. *IEEE Transactions on Knowledge and Data Engineering*, 28(3) :623–637.

[131] Zheng, Y. (2015). Trajectory data mining : an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3) :29.

[132] Zheng, Y. and Xie, X. (2011). Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1) :2.

[133] Zhou, F., De la Torre, F., and Cohn, J. F. (2010). Unsupervised discovery of facial events. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2574–2581. IEEE.

[134] Zou, Y., Xiao, J., Han, J., Wu, K., Li, Y., and Ni, L. M. (2017). Grfid : A device-free rfid-based gesture recognition system. *IEEE Transactions on Mobile Computing*, 16(2) :381–393.

Van Khoa LE

Doctorat : Optimisation et Sûreté des Systèmes

Année 2018

Détection d'évènements atypiques pour la sécurité dans les infrastructures critiques

Ce travail porte sur la surveillance des infrastructures critiques dans le cadre d'un projet nommé VIRTUALIS mené par Thales. La thèse se concentre sur la détection de comportements inhabituels dans un bâtiment. Deux approches ont été proposées à ce problème de surveillance en fonction de la qualité des données disponibles. La première approche vise à surveiller la région identifiée d'un bâtiment, généralement la zone d'importance particulière telle que salle informatique, salle prototype... Les capteurs recueillent des données qui permettent de construire un modèle statistique simple basé sur des mesures clés qui captent le comportement habituel des personnes dans un lieu. Sur la base de la distribution estimée de ces indicateurs clés, les objectifs sont de détecter des situations inhabituelles. Il a appliqué cette approche aux 2 cas d'utilisation et a montré que lorsque le but de la surveillance est clairement lié à la zone critique, la méthode fonctionne bien. La deuxième approche nécessite plus de données car elle est basée sur la trajectoire des personnes dans le bâtiment. L'idée principale est de détecter des trajectoires inhabituelles. La première étape consiste à transformer les données brutes provenant des capteurs dans un graphique d'activités plutôt simple, où les activités sont déduites des données recueillies. Ensuite, une SVM à une ou deux classes est formée en utilisant des noyaux ad hoc pour détecter une séquence inhabituelle d'activités. En mode détection, les trajectoires sont construites en temps réel sur une fenêtre temporelle donnée.

Mots clés : détection des anomalies (informatique) - vidéosurveillance - système de sécurité - machines à vecteurs de support.

Detection of Atypical Events for Security in Critical Infrastructure

This work is about critical infrastructure monitoring within a project named VIRTUALIS leaded by Thales. VIRTUALIS project is about convergence between cyber and physical security systems. The thesis concentrates on the detection of unusual behaviors in a building. Two approaches were proposed to this monitoring problem depending on the quality of the available data. The first approach intends to monitor identified region of a building, generally the area of particular importance such as computer room, prototype room… Sensors gather data that enable to build simple statistical model based on key metrics which capture the usual behavior of people in a given area. Based on the estimated distribution of these key metrics the aims is to detect unusual situations. He applied this approach on the 2 use cases and show that when the aim of monitoring is clearly related to critical area, the method performs well. The second approach necessitates more data since it is based on trajectory of people in the building. The leading idea is to detect unusual trajectories. The first step is to transform the raw data coming from sensors in a rather simple graph of activities, where activities are inferred from gathered data. Next a one or two class SVM is trained using ad-hoc kernels to detect unusual sequence of activities. In detection mode, trajectories are constructed in real time over a given time window.

Keywords: anomaly detection (computer security) - video surveillance - security systems - support vector machines.

Thèse réalisée en partenariat entre :

Ecole Doctorale "Sciences pour l'Ingénieur"