Mats Didriksen Seljeseth

UI0T-FMT: Universal format for collection and aggregation of data from smart devices

Master's thesis in Information Security
Supervisor: Muhammad Mudassar Yamin
June 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Kunnskap for en bedre verden

Mats Didriksen Seljeseth

UI0T-FMT: Universal format for collection and aggregation of data from smart devices

Master's thesis in Information Security
Supervisor: Muhammad Mudassar Yamin
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

UI0T-FMT: Universal format for collection and aggregation of data from smart devices

Mats Didriksen Seljeseth

June 2020

Acknowledgements

I would first like to express my earnest gratitude towards my supervisors Mr. **Muhammad Mudassar Yamin** and Dr. **Basel Katt** for all the help and support, that they have provided me on my project and on my thesis. Without their guidance and persistent help, this thesis would not have been possible. My greatest thanks goes out to them for all their help and for giving me an opportunity to be a part of their ongoing research towards developing a modern way of policing in a smart city.

Special thanks to Mr. **Jens-Petter Sandvik** from Kripos, that contributed with great insights into the needs of Norwegian law enforcement. This contribution formed the scope to which was used in chapter four of this dissertation. Without his contribution this chapter would not have been possible.

Lastly, a heartfelt thanks to my family and my friends who have supported me and kept me going throughout my studies.

Abstract

Information Technology (IT) has become an essential part of our lives and due to the emergence of Internet-of-Things (IoT), technology as encompassed a majority of things that humans rely on in their daily lives. However, as IT becomes more relevant in daily lives, the need for IT to serve public emergency services has become more important. However, due to the infancy status of IoT, there is a need for a data consortium that would prove to be best used in servicing policing in a technological driven society. This thesis will discuss the plausibility of creating a universal format for use in carrying out public services, such as emergency response by the police and regular law maintenance. This project will discuss what the police requires in their line-of-duty and how smart devices can be used to satisfy those needs. A data formatting framework is developed and demonstrated, with the goal of showing what can be done to unifying data from smart city sensors.

Sammendrag

Informasjons teknologi (IT) har blitt et særdeles omfattende del av menneskets hverdag. Hverdagslige apparater og dingser har blitt mer integrert med Smart Teknologi, som gjør det enklere for individet å bruke de. Med andre ord, tingenes internett har gjort hverdagen til folk flest enklere. Men i politier oppleves det lite integrering med smart teknologi, som fører til at politi tjenester ikke får benytte seg av smart teknologiens fordeler. Hensikten med denne master oppgaven er å diskutere mulighetene for å utvikle et universelt data format for bruk på apparater i tingenes internett, slik at politiet kan bruke sensor data for å utføre sine arbeidsoppgaver. Dette prosjektet vil diskutere hva politiet mangler i sitt yrke og hvordan smart teknologi kan hjelpe med å gjøre disse problemene enklere for politiet. Et data formatterings program har blitt utviklet med hensikt å vise hva som må til for å koble sammen data fra ulike smart apparater, samt hva som ikke er mulig.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Research Background

Internet of Things is arguably the most sought after topic, when discussing the emergence of technology in the modern society. In enabling seemingly mundane appliances and things, to have the capability of streamlining information over the network, a new era of information emerges. An era, in which the potential for obtaining data from items in such a way that they were not previously intended to is made feasible. Machine learning and automation in the Internet of Things has given life to the notion of using information from smart appliances to further improve how information is utilized. Data processing techniques, which were previously deemed infeasible with prior technology, has now been equipped with new methods of analyzing the data, thus making it much more affordable to do. However, the domain of the Internet of Things is exceedingly broad, making waves in almost every aspect of our day-to-day lives, from industry to our homes; the extent of research conducted on this topic is unsurprisingly tremendous. The main focus of this research is to promote the kinds of research conducted to improve the quality of intelligence gathering for law enforcement, acquisition of evidence for forensic investigations and the accumulation of useful information for sake of maintaining order in society. Since law enforcement is a broad area, comprising of a plethora of different professions, where each has their own unique set of requirements, it is crucial to establish a limitation as to what is considered relevant for contemporary law enforcement culture. According to [**Sandvik2020**] a recommended categorization of police work, to which needs can be classified accordingly, is to divide these needs into the two main categories in Norwegian law enforcement. The first category of law enforcement is chosen in accordance to the laws of the Norwegian Police Act (Politiloven). This act specifies the duties and responsibilities of law enforcement, set by the Norwegian government, to ensure peace and order in society. Whereas the second category adheres to the Norwegian General Civil Penal Code (straffeloven) and involves all post-crime activities to ensure a fair and just due process for offenders. Therefore the threshold for this research is limited to studying the application of a universal format, with adherence to the Norwegian criminal law. These two categories of law enforcement should be taken into consideration, when elaborating on its vital needs. Albeit, their place in law enforcement, namely who is responsible to enforce these two laws in society, falls upon the traditional police force and the forensic teams. All law enforcement work that takes place within a society that aims to alleviate, prevent or deter any form of criminal activity, is the primary concern of the police force. The preventative aspects of law enforcement encompass a wide collection of professions, including patrol officers, emergency dispatch units, border patrol officers, harbor

security, customs officers, immigration, traffic control officers, specialized police unit. All of which shares the common goal of being capable of preventing or intervening in illegal activities that are appropriate to their designated domain. Information and data play a vital role in combating crime in a society, and therefore these types of professions are reliant on information in order to locate and respond to incidents. On the other hand, all law enforcement work which takes place prior to the crime, which acts as a part of the legal system, to process and bring justice to society is considered goal for the forensic teams. Crimes do occur, but the events which lead up to and during the timeline to which the crime took place could spawn information. Such information would be valuable for the forensics units who aims to explain the crime in which took place. Law enforcement and forensics thrives on their ability to use information in order to perform their duties. But in a society where devices have become more prevalent, there is a new potential for improving upon the old way of policing a society. Some researchers have postulated that the traditional ways of law enforcement could benefit from obtaining data from smart devices.

A group of researchers from the Norwegian University of Science and Technology conducted a research project, as a part of a submission for a contest for Interpol, to establish the possibility of using Internet of Things as a means of improving upon the quality of law enforcement. [YSK20] proposed a semi-autonomous system that acts in a similar manner to a conventional Intrusion Detection System (IDS). It spurs many similarities with an IDS in the way that it uses data as a means of detecting when there is an anomalous event occurring on a computer system. Their proposed system, however, builds upon the idea of using the data from IoT devices to detect irregularities in a smart city, rather than on a computer system. The main difference between using a collective of devices in a city or a building as a starting point for an incident detection system, as opposed to how IDS' perform the same procedure on a computer can be explained with the following: Imagine how incidents are detected in a regular IDS, where the behavior on the computer (i.e. files are created, registry modified, ports opened) and network traffic (connecting to a specific domain, specific traffic sent) would be what triggers an alarm. In [YSK20], any form of data captured on a IoT sensor could be treated in the same manner as an IDS. Drastic changes in temperature, human presence in a home where all inhabitants are absent, a fluctuation in heart rate from a health monitor can all be considered as "behavior" in the same way an IDS watches its computers behavior. In other words, the proposed research is the same as taking the idea of Intrusion Detection and applying it to all functions that makes up society. Behavior in this sense is the data that a sensor records and this is what researchers are trying to leverage to improve societal functions.

The use of camera surveillance is by any means considered to be a conventional example, of using technology as a means of reconnaissance. However, the normal way in which a visual surveillance is conducted, is done in such a manner that it relies on the attention of a human operator. A method of surveillance in which their the operator's personal capabilities, sets the limitation as to whether an incident is detected or not. However, with modern computer vision

technology that is paired with the use of machine learning, could eliminate the requirement for a human moderator. And this could allow for a more comprehensive coverage of more information simultaneously. Project Rocket is one of the most recent instances of a project which has introduced the Live Video Analytic paradigm, where neural networks can process video feeds as they arrive on an endpoint. There are a multitude of paper published, using this solution to solve challenges in our society. For instance, in [Ana+19] live video analytic were used to develop a safer way of implementing a crosswalk for people with disabilities. Camera feeds were in this case used to determine whether the crosswalk duration required a time extension, whenever a wheelchair user was to cross a road. Another example of how IoT can be used to leverage the prediction and alerting of events can be seen in [Jun+20], where the research group provided a means to which IoT devices are used to predict the likelihood of a natural disaster.

1.2 Research Questions

With such a novel concept being developed using a modern technological landscape, to enhance the capabilities of law enforcement, there are a few issues that must be addressed in order to progress in this area. This research will focus on four main problems relating to how a city wide policing system is to be developed. As a starting point for creating the format for a smart policing platform the following concerns must be addressed.

1. *What type of information would be necessary for law enforcement agencies to carry out their tasks in a proactive manner?*
2. *What category of IoT devices in a Smart City can be leveraged by law enforcement to obtain useful knowledge of on-going crimes?*
3. *What data format structure of high-level data will ensure that data output can be processed from different devices and outputted in a unified format?*
4. *What are the legal and ethical considerations that must be addressed, in order to preserve the individual's right to privacy?*

First of all, the current status of law enforcement has to be addressed with focus on the compulsory needs, that is to establish: what the law enforcement requires in order to do their respective duties? As police often do meet resistance on duty and has to encounter situations that deemed unpredictable and hazardous, it is essential to address what type of requirements a policing system must meet in order to supply a forensics team and law enforcement with essential information. This issues has to be addressed this way, because it is not abundantly clear as to what the law enforcement requires from a smart policing system in order to do their tasks better. Thereby, when answering this question it would become more clear as to what requirements must be attended to when developing a smart policing system. Internet of Things has gained a wide grasp on our society. Devices and sensors are put in place in nearly all aspects of out world and therefore it may be unclear as to what potential could be exploited by a policing system. This question aims to raise the concerns with the diversity of devices that exists today, which of those are useful for policing and what data could law enforcement gain from them. This questions serves as a guide to which the most prominent devices that exist today and how they are being used by law enforcement, can further help fighting crime. The core idea of this project is to develop a form of implementation for a unified format to which police and forensics could stand to benefit from. It is considered a compulsory concern in this research to be able to demonstrate the novelty and test the feasibility of a proposed unified format. It is done in such a manner to convey the core ideas of why the format exists and how it is planned to be developed in a real-world scenario. One primary importance of developing the format for a crime related instance, is to find the best possible way in which the data can supply the officers with the knowledge they need to be able to respond and hypothesize more efficiently. The last question is added, granted that the proposed system, as it primarily builds upon the ideas described in [YSK20], to develop a piece of a system that collects a mass amount of data.

One of the major concerns with such a system, as it is similar to surveillance in some areas, there are obvious ethical, legal and security concerns that must be declared and addressed. All of these aspects will be discussed in the final chapter to deliberate on what is considered crucial for one to take into consideration, when developing any system that leverages data from any devices.

1.3 Current issue

Traditional police work has been practiced for a long period of time. Outdated phone emergency services, high fatality rates in patrols and the rise in deaths of alleged suspects are some of the realities in modern police practices. With new technology that has the potential to give police more information about what is happening in smart city, the question that one have is: How and in what way could law enforcement benefit from IoT and smart sensors?

1.4 Reason for chosen methodology

Design science Research that was introduced in [VK04] is the chosen methodology for this thesis. It is a circular type of research methodology, where the researcher identifies, defines a problem and builds a solution to solve the identified problem. Once the solution has been developed the next step is to demonstrate and evaluate the artefact that is the proposed solution, and at the end the result and are communicated, such that the process can start over and improvements to the artefact can be made. The reason why DSR is an ideal methodology for this research, is that the development of software will always require further improvements to be made. Software is never fully developed on the first iteration, there is always ways that is can be built better. This methodology is therefore ideal to address how software could be a solution, and how its shortcomings can be improved upon.

1.5 Thesis composition

The following chapters in this thesis will be focusing on the past solutions proposed in this research, but the main core of this thesis will be focusing on is the answering of the four research questions. In chapter two the main focus is to discuss what has been attempted in past research with respect to this topic. Chapter three will be discussing how the research questions appeals to the components of the Design Science Research methodology and how each of the questions are attempted to be answered. The Fourth chapter is the first question, where the needs of police in a smart city will be explored. Devices and how they can be used to inform law enforcement is the main topic in chapter five. While a proposal for a solution and its limitations is the main concerns in the proceeding chapter. Problems faced by implementing the proposed software solution in chapter seven will be discussed in the proceeding chapter and an evaluation will make the eight chapter of this research. A discussion of all prior core topics followed by a conclusion are the final two chapters of this thesis.

2 Related Work

In the area of research on the unification of smart device and smart sensor data, there are many research papers written. Given that the Internet of Things is a recent phenomenon, the degree to which research has further progressed this phenomenon is extensive. However, given that the age of IoT is still recent, the amount to which the standardization of some aspects remains untouched. This can be seen in the vast number of devices that exist today, manufactured by small and large companies, who all sought to gain a market value. In achieving so, a lot of proposed standards have been published alongside the devices that merged during rise in popularity of IoT devices. Where one device might use a wireless protocol such a Zigbee, while another similar device might use a protocol like Z-Wave or Bluetooth Low-energy (BLE). Another issue that arose during the IoT-boom was the disagreement revolving around what the devices were to communicate between each other, thereby creating a diverse pool of data formats. Formats that spans along various byte-orders and serializations, which makes the idea of interoperability difficult. In order to develop a system in which a police force is to collect data from smart devices, a standardization must be established.

For the sake of developing a data unification framework in pursuit of crime monitoring, the amount of research is sparse. Mainly the research activities that revolve around the development of a unified framework is more focused on the developing such a framework for commercial and industrial use. For instance, in [NB17] a novel data aggregation model was proposed for use on environmental sensors in a smart city. The focus of their research encompassed the use of network sensor data from industrial-based sensors, which monitors water, electricity and gas-based sensors. A more concrete implementation of a model that can be used to translate smart city sensor data was proposed in [Pra+18], where they demonstrate how the North Atlantic Treaty Organization (NATO) can leverage smart city appliances in urban operations. In disaster situations that takes place within an urban environment, the research group believe that the integration between IoT devices in a smart city environment could be integrated with the systems in a Federated Mission Network (FMN). This integration was proposed to aid soldiers in obtaining intelligence for use in their vehicle systems, urban personnel deployment and UAV systems. What makes this paper relevant for this project is that it outlines a few similar ideas that are aligned with the notion promoted in this research: to utilize peripheral data sources in pursuit of a safer execution of tasks in an urban environment. In [Jun+17] a novel way of integrating crime incidents and police vehicle locations in a smart city was proposed. This system would utilize smart technology in conjunction with police vehicles in order to supply its officers with information about the crime in advance. It is not strictly a research into the use of smart city data, but the paper does illustrate a good point on how technology is utilized to

effectively dispatch units. Moreover, the project also discusses a relevant point in regard to the use of GIS technology to improve the logistics of dispatching units. Whereas the city is divided into areas, where the crime model bases itself on the location where crime is the most frequent. This type of approach to distributing crime events based on the location where they occurred, could be an ideal way of establishing logistics for a smart policing system.

[Gho+16] explored the prospect of utilizing data mining techniques on heterogeneous data to provide law enforcement with a bigger picture of the incidents, that takes place within the city of Newark (New Jersey). This paper clearly focuses more on integrated cooperation between precincts and the use of government registries to improve crime fighting, but their ideas still shares the same sentiments towards the use of data mining to aid in the process of ensuring public safety. Further elaboration on the potential for data fusion of smart city data is discussed in [Din+19], where a comprehensive survey is conducted into research conducted on the topic of IoT and Data Fusion. As a whole, this paper provides an adequate overview over the requirements for fusion of data in a Smart city environment. Furthermore, the article also defines data from common devices found in a smart grid, supplied with information pertaining to the category of the data. [Wan+15] covers several important concepts in correlation to the Internet of Things and assembling data into a new set that can provide the consumers with more information. In conjunction with data fusion concepts, this article focuses on ideas, such as the construction of incidents, based on the readings gathered from smart city devices. Moreover, they focus also on context awareness and its importance in IoT to supply more information about events, using multiple data sources. As a result of this survey, they propose an evaluation framework for data fusion with 10 points covering their core topics.

Another unified format for smart sensors was proposed in [Cen+14], where the authors translated data from protocols such as CoAP, 6LowPAN, UDP, 802.15.4 to detect anomalies in smart city sensors. Location tracking technologies were used in conjunction with data from these sensors, to detect anomalous readings in humidity, light, Carbon Dioxide (CO2) levels. Furthermore, the data were placed in a timeline using timestamps from the data entries, such that these data metrics were properly mapped by time. A linked data formatting for various smart city devices were introduced in [BAA17]. Their proposed framework consists of a framework that scrapes IoT sensor data and stores it for a user to semantically link it to other data. Resource Description Framework is utilized in their framework to link data together. The core point made in this article is that raw data from sensors can be transformed into meaningful data and assigned semantics that could aid in identifying and classifying the data. By assigning meaningful semantics to a data source would be useful in a situation where an area has a large pool of devices, where they need to be distinguishable from one and other. Furthermore, [Lau+19] discusses the prospect on the establishment of an ontology framework for data fusion of smart devices. Their proposed framework comprises of six layers, which covers the various aspects of conducting data fusion of smart devices to a significant degree of depth. Their research introduces an intricate list of attributes that makes up these six perspectives, that

are used to fuse data together. Ranging from the scale of the fusions, to the objective of fusion, the means to which data is communicated, the output type and so on. This research paper, in essence, poses an important role in this research when developing a way to fuse data together, as it covers an adequate range of attributes that has to be taken into consideration. Moreover, the appropriation of such ideas, posed in this research, also fits the narrative of a smart city with its proposed perspectives.

Another system that has similar ideas in mind, would be the data fusion model that were introduced in [Jar+18]. In similar manner to [Lau+19], this research also categorizes data based on similar attributes. Main differences between the former and the latter article is that their proposed testbed for experiments is done on a Software Defined System environment. It is done in such a manner to reduce the control complexity of sensors, which is seen in how they differ slightly from each other in data assembly and communication. Further testing of their proposed data fusion model was done on a multitude of Air Quality sensors to fuse this data to calculate the Air Quality Index. As a side note, the main focus of this research is motivated by the potential of improving the quality of the medical field, as opposed to a general coverage of society or a motivation that adheres to the needs of law enforcement. A four-step model was proposed in [Gau+15], where the intention was to create a model which could serve as a recipe for creating a common format. The levels proposed in this paper consist of a collection phase, a processing phase, integration phase and an aggregation phase. It is also suggested that the formatting for the data should be done using semantic web technologies to transform the data into a RDF format. [Sha+17] demonstrated a different way in assembling a format for various devices. A format based on JavaScript Object Notation (JSON) was proposed, where it was demonstrated on raw data that was generated by streetlights. It was demonstrated data templates for streetlights comprise of messages, as well as identifying information about the location of the lamps. A novel framework for structuring data was also proposed in [Kol+19], where the research team proposed the Stream Annotation Ontology model (SAO). This data format was introduced as a part of the smart city project CityPulse and illustrated a different way of structuring time-series data from open data repositories that is supplied by the city of Aarhus in Denmark. As noted in the paper, the main focus of city pulse is the use of open traffic events with timestamps and location data to form a bigger picture of incidents that takes place in Aarhus. A formatting standard for use in sharing evidence artefacts that originates from IoT devices were introduced in [ZCB19]. Their data format was presented as a means to which law enforcement could share their findings and experiences from working with IoT devices. Law enforcement lacks the ability to efficiently share evidence and experiences from working with IoT devices. Therefore, [ZCB19] developed this format for the sole purpose of allowing LEAs to share this information in a manner that does not reveal sensitive information and also allows for an easier way of reading the accumulated sensor data.

3 Methodology

3.1 Research methodology: design science research

While Design Science Research (DSR) is a relatively new proposed methodology of conducting research, it allows for an alternative way of approaching a solution to a problem. This research methodology from [VK04] has been accepted in recent years as an appropriate research methodology for Information Systems (IS). [Pef+07] proposed that the Design Science Research methodology should consist of the six following steps:

1. Identification
2. Definition
3. Design/development
4. Demonstration
5. Evaluation
6. Conclusion

This dissertation attempts to address 4 primary research problems. These problems fits into this methodology, where the goal is to develop a solution to an underlying problem that is persisting in our world. In chronological order from first to last question, these questions attempt to cover the steps in the DSR methodology. In order to identify the issue with law enforcement and what they lack in today's world, where smart technology has a significant presence. Question 1 in this thesis aims to identify what law enforcement could expect from IoT. A combination of declaring law enforcement's core duties alongside a discussion on the problems that they encounter is the main contents of the first research question. This scope of this question will be limited to address the needs in Norwegian Law Enforcement.

To define the ramifications and possibilities of this field, research question 2 discusses what type of devices exist in IoT and which are possible to install in a city. With this question, a first threshold is set for this project. To declare a list with relevant devices in a smart city, the context as to what is achievable with smart technology in law enforcement becomes clearer. However, the limitations of smart devices and its data also needs to be declared.

In research question three, where the artefact is to be discussed, the introductory sections will attempt to discuss the limitations with this project, in relation to the problem of non-conformity in IoT data. For the remainder of this chapter, however, the aim is to provide details of the proposed solution. A documentation of the proposal is provided, alongside a coverage of the artefact's core functionalities. An evaluation is provided to test the proposed format against other competing standards. This will be in chapter and aims to discuss how the program compares against other application level protocols. The two metrics that is going to be measured

are reliability and performance.

Ethical and legal concerns are explored in further detail, to discuss whether this proposed method has any negative impacts to society. The ethical and legal problems will be discussed using Security by Consensus model from [Kow94] to explain how the various aspect of the program can holistically affect each other.

3.2 Problem 1: Identifying Law enforcement needs

It was the famous American author and Salesman Zig Ziglar, who once said that:

> "The first step in solving a problem is to recognize that it does exist."

This statement holds true for solving any type of problem, and especially when assessing a core problem for a research activity. To nurture the core philosophy of this dissertation, the first required step is to establish the definition of the problem. After defining what the problem is, only then does it become clear as to what the proposed solution should be. The problem that is going to be addressed here is purely related to the Law Enforcement profession and their needs. More specifically, how could they benefit in their line of work with additional information at their disposal. In order to answer what the law enforcement personnel requires, a combination of reviewing former literature on the subject area. Additionally, the use of authoritative figures in the field were questioned with the intent of probing for a general point-of-view on the matter.

The questioning were done over e-mail to a candidate that were considered an authority on the subject and would therefore be able to provide an adequate degree of insights on the subject matter of law enforcement. While it is courteous to declare that Mr. Sandvik has been involved in the project together with [YSK20], the answers would not be biased in any form, as their answers would come from an experienced point-of-view, rather than a biased one. The questions that were asked during the communication with Mr. Sandvik involved questions regarding the basic needs of police, asking for a professional opinion on the matter at hand. The aim was not to probe for very specific answers, but rather for an essential coverage of the research problem. As a supplementary source of information for further answering the first research question, a literature survey was conducted to probe for more specific needs of law enforcement. Specific search terms, that relates to problems relating to law enforcement were used to procure articles that were discussing the topic of police needs. Terms, such as, *situational awareness*, *deterrence*, *crime prevention*, were used to find relevant material to further elaborate on this research question.

3.3 Problem 2: Defining relevant devices

While identifying a problem is the first step of being able to solve it, the next phase is to set the framework for how it is to be solved. This could be done in the form of exploring possible solutions that exist today and further develop it to fit current issue. One particular limitation that should be set in relation to this research is what devices are considered relevant. Because

there are a vast landscape of devices in a city, it becomes difficult to assess all areas of the city of all potential devices. Therefore, a formal definition must be defined of what can be achieved with which smart devices. A literature survey was used to find relevant material for establishing what type of smart devices do exist today. The use of articles and diagrams that list specific category of devices were the targeted material. Search terms that relates to forensics and law enforcement, were used together with the types of devices to find relevant information about the device and how it would be useful for law enforcement. The intention of this method was to first list the devices that could potentially be of use and then go into detail about how the device would be used in a forensic investigation or in a law enforcement scenario. For instance, upon discovering that smart watches possessing the capabilities of health monitoring, the next step would be to search for articles regarding forensics investigations on health tracking.

3.4 Problem 3: Developing, designing and demonstrating the solution

This question is a bit different as it pertains to a specific area of study, that requires some testing and discussion around the best practice of assembling data in a format that would be considered optimal for law enforcement. The third question would involve the discussion of what type of data is considered to be of relevance, as well as how it should be structured. Mainly this will take place in the development phase of the solution, where eventual disadvantages are addressed. The reason why the relevancy of the data is important is that the proposed format requires that an overview over the limitations are declared, before the artefact is presented.

3.4.1 Development

A demonstration would be made for the intents and purposes of showing how different devices can be connected together and be handled by one and same program, to provide the operator with an output, based on the devices that are connected to the software. The testing environment comprise of three main parts, which are all controlled by software written in Google's Golang (Go). One program is written to handle the data that the devices are generating, while the other two are examples of data sources, whereas one of whom is a realistic hardware-based application, and the other is a virtual software simulation. The testing devices used in this section is a combination of hardware based solutions. Which in this instance is an LoRa MKR 1300 Arduino based board, that is connected to a cloud-based service through an in-house LoRa gateway from Mikrotik. The approach to the hardware-based simulation, is done in the following steps.

First the arduino board is connected to a temperature sensor, that is feeding information directly to the embedded board, which is then configured to assemble the recorded data into a packet. Second step is to dispatch the packet on-wards onto the cloud service for storage and this is done by broadcasting the message over a 800MHz frequency, which will be picked up by the local Mikrotik Gateway. Third step is that the gateway is set to receive any message that are broadcasted on the 800MHz frequency range and then send those messages onto the cloud. Fourth step takes place on the cloud, where the message is received and stored. After a message

is stored there a third-party application is used to retrieve the uplink messages and the messages can then be formatted. Due to the lack of diversity in the type of sensors available at the time of developing the software, a software-based solution were implemented to simulate the emerging devices available in a smart home. The software is written in go as well and simulates step 1-3 in the hardware-based implementation, although the data is artificially generated and managed through software. One major difference from how the hardware demonstration is being performed, is that the software simulation possesses more flexibility in its features. For instance, it can be written to support different uplink data formats. This would allow for a more diverse range of data sources, which can be added to the linking of the developers. In order to collect the data from both the hardware and the software simulations, a separate piece of software is created. It has the capabilities of reading uplink data and manage existing devices that are connected to the Things Network. Furthermore, information relating to how this program is operating should also be briefly discussed, such that it can be reproduced by others for the sake of future developments. This is especially a crucial detail to discuss, granted that some aspects of operating the program is not clear to the reader. As such, it will aid the reader/researcher to understand how to use the program. More details of this is contained in the appendices.

3.4.2 Demonstration

At this stage, the two implementations are going to be tested on the formatting side of the program (SmartPolicing-Interface) to show that the program is capable of obtaining information from simulated and generated data streams. In the demonstration, the intent is to show how one could configure a formatting recipe for how data should be interpreted by the go program.

Generated data will take place over an Arduino instance, where the humidity and temperature readings are provided through an actual sensor. This sensor is located in-house and provides periodic updates to the Things Network (TTN) cloud. The other set of devices are being artificially generated as a means to simulate data from devices that are otherwise difficult to simulate in a proper hardware-based manner. Sensor data, such as hue color values, door lock state and so on are generated through a Go application. All devices generated through the software based simulation is done using regular Websockets.

3.4.3 Evaluation

Lastly, the results from the demonstration must be evaluated in order to draw out the strengths and weaknesses of the project work. Using other similar projects as a baseline for the evaluation of the universal format, the comparison of the proposed method and the competing formats are to be assessed. The criteria for assessing the framework will be based on two main metrics to evaluate the underlying application layer protocols of this program in comparison to other standards used by the competing frameworks. The first metric that will be evaluated is the performance of the program. This metric is intended to show that the program performs well and that it can be put to the same standards as the competitors. The second metric is to test

the reliability of the program. By reliability in this case, the idea is to test the networking and processing capabilities of the program with respect to packet loss, and how performance will affect his and vice versa.

3.5 Problem 4: Legal, ethical and security concerns

As with many projects that set out to propose a solution to a problem that we are facing today, there is the possibility, that with this solution, comes new problems. Mostly in the form of legal and ethical ramifications, which arise from the existence of this solution or from the way it is developed. It is by that very fact, that a chapter must be written to address any complications that could arise as a result of this project. The last research question in this dissertation will set out to address the inevitable issues, mainly the legal framework and the ethical concerns relating to it. Both of which is by all means a necessity for this project to adhere to and satisfy. For instance, the use of data must be done in accordance with current legal regulations, such as EU's GDPR. The GDPR, amongst other legal literature, must be analyzed to assess the means to which this project manages all the data that is procures. On the other hand, the ethical standards that exist today also governs over how data is being used by humans and machines. It is thereby as pertinent as discussing the legal ramifications, to also address how a component of a semi-autonomous crime detection/prevention system poses risk to an individual's right to privacy. Questions in regard to concerns about this project being a part of a surveillance system is also discussed. Software security is the last sub-topic that makes up this chapter. Since this project handles data that is being generated from all parts of a city, there is a probability that this data will become a target for criminals to obstruct or illegally obtain. The security aspect of a written program is a vital topic to mention, as the nature of this project is to handle data from a massive array of devices. General software security concepts, as described by OSWASP will be discussed to emphasize which of standards of software security is applicable to this type of project. The Security by Consensus model that was proposed in [Kow94] is used as a means to explain the important problems discussed in this chapter and how they inter-relate and depend on each other.

4 Police needs

As described by [**Sandvik+20**], Law Enforcement can be divided into two groups: first is the group that aims to deter, alleviate and prevent crime in society and there is the group who is responsible for ensuring that the law is practiced in a just manner. These two groups makes up the core responsibilities of a Law Enforcement Agency in Norway. The first of which is referred to a patrols or field-operators, as this group consist of the men and women that are actively in the field, fighting and preventing crime as they occur. The second group will be referred to as forensics, since their responsibilities are to procure and analyze data as they are discovered at the scene of the crime.

4.1 Field-operator's needs

A field operator is the active force of men and women, who are out in the cities, suburbs and rural areas to Protect citizens from harm and loss of property. Their daily routines consist of unpredictable encounters and situations, that puts these operator's lives in danger to ensure that other's lives are protected.

4.1.1 Intelligence

The use of intelligence is arguably not a new practice in risk-related professions. In activities, such as wildlife preservation, military operations, geological excavations and so on, there is an element of prior knowledge, obtained by its participants, in light of executing their respective tasks. Obviously, the tasks being carried out in this manner, is done so, to reduce a reaction from occurring as a result of the work. It is considered in some way, a risk assessment of the underlying and/or peripheral environment, in order to arrive at the conclusion, on whether the ensuing task/challenge will cause an unwanted event to occur. Military work often uses the term intelligence as a term for information obtained on relevant adversaries that could aid in gaining a significant advantage over said contenders. In geology, the use of intelligence could be in the form of a conducted risk assessment of a geological site, to assess the likelihood of there being hazards being present [YHJ18]. With no regard to the nature of the profession, whoever utilizes any form of intelligence, the general census for using such methods, is to improve upon the quality of the job-outcome, or purportedly to reduce the likelihood of accidents and injuries. With respect to the police profession, the use of intelligence is postulated to be a deciding factor, which could affect the outcome of an officer's work day, as well as their prowess. Use of intelligence is doubtfully a novel concept for law enforcement either; given that use of information to point patrols in the right direction has been practiced for decades with the emergence of telecommunication, such as seen in the use of emergency hotlines.

[IK19] inferred that the needs of the police are to provide its operators with the necessary information, from the environment to which they are operating within. For instance, with the use of old surveillance technology, such as Close Circuit Television (CCTV) systems, can provide operators with the necessary information to adapt to a difficult situation. For instance, the deployment of a comprehensive CCTV coverage over a metropolitan area, could yield a beneficial opportunity to law enforcement, as it would allow information to be fed to its operators directly. Such that a bigger picture of the situation, to which they are about to interfere is provided, prior to the engagement. It is further proposed in [ELD19] that the main issue in policing is the lack of situational awareness (SA) and thereby introducing more task load and stress to its workers. Their research proposes the use of location services, alongside sensors which provide a crime feed to its operators through Smartphones and smartwatches. This research concluded, however, that the stress-level that were attributed to low degree of situational awareness in patrols, were unaffected after a higher level of awareness were stimulated with the new proposed system. Without the proper intelligence present in a law enforcement grasp, the ensuing situation will become less predictable and thereby this could lead to injuries and the psychological issues.

Police officers, much like soldiers on a battlefield, have to expect encountering volatile situations, to which they have no foresight over the eventual outcome. Regardless of the situation involving the forced entry into a home, with no information about what the officers are to expect on the other side of the door. Neither would an office be guaranteed to be in full control, in a bank robbery scenario, where the outcome can branch in multiple directions, depending on every action taken during that event. With hazardous situations that could last anywhere from a few minutes to hours or days, the lack of control in these situations, could lead to personnel sustaining injuries. [Tie+18] conducted a study on a decade of records pertaining to all non-fatal injuries inflicted to U.S based field operators between the year 2003 to 2014. According to this study, the overall numbers of non-fatal injuries reported reached 600 thousand individual reports from police officers working in the United States. According to [Lyo+17] the number of non-fatal injuries reported by law enforcement personnel ranged from 240 to 2500 injures per 1000 personnel. Most common form of injuries sustained were Musculoskeletal type injuries in the upper extremities. More specifically, reported types of injuries were strains and soft-tissue sprains. Injuries are a relatively common hazard for law enforcement personnel in the field, however, the lack of control in an altercation could also lead to death(s). Physical harm is a persistent problem for field-operators, but there is a less visible problem; the threats to an operator's mental health.

[Pri17] discusses the psychiatric difficulties that law enforcement personnel faces on their job, whereas, they hypothesize that a good portion of the unsettling and unpredictable incidents that one may encounter, will inevitably degrade their overall mental health. Rooted in the events that transpires over the course of a work shift for law enforcement personnel, there are a possibility that some of these events will affect their experiences at work. Limited by

the mental strength and the character of the individual personnel, the experiences pose a significant risk to the overall quality of their psychological health. If the individuals, to which these unsettling events are exposed, does not have the capacity to process such events, the risk of developing further mental conditions are more probable. According to [Pri17], one of the major factors that attributes to the decline in mental health of law enforcement personnel is attributed to psychologically traumatizing experiences, such as being physically injured, losing co-workers on the job or developing nervousness as a result of repeated encounters with unpredictable situations. [Jen19] conducted a qualitative study on the prolonged exposure to traumatic events, using Ehlers' and Clark's cognitive model from [EC00], to discuss the likelihood of predicting the likelihood of detecting signs of Post-Traumatic Stress Disorder (PTSD). In the results of this study, it was identified a correlation between negative appraisal and exposure to negative situations and how it would have negative psychological effects on the Law enforcement officers. Effects, which in this study was reminiscent with those symptoms associated with PTSD. Without the necessary functions in place to provide field operators with the appropriate intelligence of a situation, the likelihood of avoiding traumatizing events is reduced. Primarily due to the fact, that the information needed to be able to adapt to the situation and act accordingly is not available, and therefore the probability of miss-step is inevitable.

4.1.2 Deterrence

While the previous sections of this chapter have discussed the possibilities of lifting the quality of police work by reducing risks and improving information flow, there are other potentials which can be exploited in a smart city environment. This part does not include the quality of the information that could improve the police officer's work efficiency and safety, but rather reduce the number of crimes to which they are required to respond to. In order to reduce the number for crimes within a metropolitan environment, there must be an element present that discourage or deter the perpetrator. In a smart city there could be a potential for employing various types of devices, which possesses the capability of warding off potential criminals from carrying out their heinous acts, by simply being installed and present. In other words, some smart devices, when configured appropriately, can serve the public by acting as a means of which the crime does not take place, due to the possibility of the crime being recorded by the device's sensors. There exist at this time various implementations of IoT devices and ICT devices which can have this effect on the general public, by simply being present and visible to the public. To categorize the types of deterrence effects that a device can have on an ongoing crime can be divided into two primary categories: passive deterrence and active prevention. In the former category, the idea is to install and promote a smart system that could be place within a residential or a public space to serve as a guard against non-compliant citizens. In order to comprehend the strengths of utilizing passive protection within a public space, consider the following scenario:

The Chinese government saw an increase in success of enforcing their policies and maintaining

order through the use of a new social credit system. This social credit system imposes an individual score for each of the Chinese citizens, whereas the higher this score is, the more that citizen would have opportunities in their life. For instance, an individual whose score is considered high would be eligible to enjoy financial and lifestyle benefits, which the Chinese society is mandated to allot to upstanding citizens. Some of these benefits include, but are not limited to, higher chance of receiving a loan, more likely to be hired for a job, higher chance of being able to rent an apartment. On the other hand, a more severe effect that rose in light of this social credit system, is the social pressure to "fall-in-line" with the societal norms, where outliers (with low scores) are ostracized from social groups with higher scores, because the credit system penalize associating with those of lower values. This form of passive governing forces the mind of the individual to "*fall in line*" and obey the law, by impeding social and financial pressures upon its citizens. Its nature becomes passive, by how it acts to prevent an action from being carried out, because the individuals are unwilling to take the risk of penalization. It is simply put, a form of discouragement put in place to enforce a policy or law upon a group of people, without the need for physical presence to enforce it[**Backer2018**]. Another example of passive deterrence was introduced in [Gal19] where the team demonstrated the de-escalate project in the clubbing district Stratumseid in the Netherlands. The general idea of this project was to install a CCTV camera, that was equipped with the capability of detecting and recognizing human faces. The purpose of this installation was to detect the faeces of potential assailants, due to the rise in amount of physical altercations that was occurring in this particular district. What the research team in [Gal19] discovered what that the presence of the camera, caused a reduction of violent crimes in that area. They postulated that the reduction in violent crimes were mainly attributed to the psychological effects of being under watch of the police. In [Piz+19] it was further inferred that the process of deploying surveillance cameras in a city would cause the individual to feel that they are being watched by a omnipresent entity, thus the likelihood of a crime being committed is thereby reduced.

In a similar manner to passive deterrence, the effect of discouragement of committing crimes can be actively enforced through the use of technology, but in a stark contrast to passive deterrence, the technology is more present and active in preventing the crime. One particular instance in which the Chinese utilize an active approach to resolving the societal problem of jaywalking is to publicly display the perpetrator to the public and shame them. In stark contrast to having a potential of reducing one's social credit score, this form of penalty involves a more active element of public shame, where the offender is publicly displayed on a large screen for everyone to see. Being able to respond to a situation is not entirely left to the emergency response and field patrols, but also on the individuals who are witnessing or are victims of a crime. In any regards to the accuracy and timeliness in reporting a crime, the specificity of the reporting must be appropriate to ensure that the police is able to respond. With IoT data, this could enable for a timelier reporting of events occurring in a city, without the need for a human element to report it to the police, which means that the field operator's ability to

prevent a crime will improve drastically. For example, the use of sensors in parking lots has provided traffic police with a more improved way of scouting for parking violations. This is an improvement in comparison to how parking violations were originally cited by traffic police, where the operator has to be present to be able to report the violation and file a citation to the perpetrator.

4.2 Forensic needs

Digital forensics is another aspect of law enforcement which could benefit from utilizing data from smart devices to investigate a crime. While digital forensics are a well established field of law enforcement, there are other unexploited data sources which could be used to further infer the extent to which a suspect is guilty. What sets forensics apart from patrol-duty is the fact that the work undertaken by forensics staff is oriented towards answering the questions left behind, prior to the completion of the crime. Police patrols main tasks is the prevent and intervene in an ongoing crime and does so through direct intervention, rehabilitation[1] and deterrence. A crucial distinction must be made when addressing the need of law enforcement, as these two working-groups require disparate requirements to do their job. Forensics are an important part of law enforcement which enables the juridical system to fill in the blank areas, that is the untold stories of a crime, which could play a crucial role in providing the evidence that would enable the law to be served to the best of its ability. Forensics are a significant element in juridical process, where forensic analysts conduct investigation of a crime scene. Analysts in forensics process would observe the scene of the crime for the evidence which could fill in the blanks of a criminal case, such that the appropriate legal action can be taken against the perpetrator. All work processes in a forensic investigation differs from how the field related duties are laid out, meaning that the main focus is not on being able to apprehend the suspect safely, nor is it to be able to prevent a crime. Rather the main focus of a forensic analyst is to be able to provide evidence that could aid in an investigation, such that the suspect(s) are apprehended and punished. In order to achieve this goal a forensic investigator has to examine the evidence that relates to the case and in order to do so they have to acquire such evidence.

4.2.1 Evidence acquisition

Regardless of what is considered important to a forensic analyst, in contrast to a police officer, the data found on a IoT device, still pose a significant value to the analyst. Regular digital forensic evidence, meaning what is considered evidence found on computers and mobile devices, can be strengthened or weakened by introducing evidence that originates from IoT devices. However, as IoT is not an established means to which evidence is procured by investigators, a couple of challenges emerges when IoT data is to be acquired for forensic analysis. One of these challenges is on the unfamiliarity and lack of methods to extract useful information. According to [MBS18] the footing of forensic methodology in the IoT landscape is lacking. Stating that as more devices are connected to the internet and being able to transmit data

[1] Extent to which rehabilitation of criminals, depends on a country's legal system

over it, the challenges of acquiring and handling the data for potential evidence will worsen. [MBS18] further lists the three primary categories in which data is stored in IoT. The first one being the smart sensors, the second one is the intermediate connecting devices including hubs, computers and routers and lastly cloud platforms that aggregates and handles sensor data. All three categories pose challenges to evidence acquisition due to different issues.

Data acquisition is the primary concern when attempting to address challenges in relation to carrying out forensic investigations in IoT. When approaching the crime scene at a *physical location*, the challenge of obtaining any useful evidence from a physical sensor becomes a challenge because of the data being stored in volatile storage. [**Sutherland2017**] illustrated an example of this issue from experience in working in law enforcement. A forensic case that took place in a smart home, were held due to an accident. An employee, that was attempting to gain physical access to an IoT sensor accidentally tugged the wrong cable and cut off the power to the device. As a result, no data could be retrieved, because the nature of volatile data is set to destroy itself as soon as the access to power is gone. [HV17] further states that there are insufficient coverage appropriate methods of acquisition from these devices. This is mainly attributed to the diversification of implementation of hardware, software and communication protocols. In other words, given that devices are not created equal, there is little potential to directly access the data because of how data is formed prior to aggregation and storage. In [Gar10] the development of tools to adapt to the changing landscape of ICT moves towards a outcome where new tools developed for forensics does away with the former tools. Mainly it is referred to by the authors as "*tactical reverse engineering*" approaches to adapt to these new emerging technologies. As a result, the current research method fails to adapt to the old and leaves no room for developing established standard methodologies for further research. Much like this prediction, this statement holds true for forensics when it comes to researching investigation methods of smart devices. Devices are different from one and other, and thereby the research becomes on how to adapt to a one or a small set of devices, thus limiting that research activities reach to its constrained set of devices. Data stored on sensor devices must be extracted without altering the state of the device that could obstruct the availability of the evidence. Diversity is the second major setback from direct access to these embedded systems. When evidence cannot be retrieved directly off of a device that is physically present at the crime-scene, the next step is to attempt to obtain the evidence from an intermediate medium.

In IoT the *intermediate layer* represents the devices that acts as a connection between smart sensors and the cloud. For devices and sensors who communicates wireless, through communication protocols such as 6LowPAN, Bluetooth, Z-wave, Zigbee, 802.15.4, LTE or LoRA, have to connect to an intermediary element before the data can reach the cloud. For the devices who relies on wired connection the case is the same, as they would also require some sort of connecting relay. [Mef+17] Some devices that are a connecting medium includes, conventional computers, smartphones, proprietary smart hubs, routers and gateways. When it comes to intermediate level, the devices including phones, computers can be approached using conventional computer

forensics. This is due to the fact that research into investigating these instances of hardware is well documented and commonly practiced. When obtaining access to these devices, the only requirement is that the credentials to the controller is available. However, with the inclusion of mediums, such as gateways, smart hubs and routers, the availability of the evidence relies on whether or not the manufacturer allows its users to access their hardware. In [Jan19] some devices that communicates with a hub over wireless protocols can be intercepted, using special hardware that communicates with the same protocol as the device in question. However, depending on the device, this data could be encrypted. In some cases, the data is not encrypted at all. How the data is assembled on the device before being sent to the hub is another problem with device to intermediary interception. If the data is assembled in a way that is not understandable, then additional steps are required to reverse engineer and implement a decoder for every device who sends data this way. According to Samsung's SmartThings documentation [Sma20] on Zigbee, illustrates how a zigbee Join request is assembled. At first glance, their implementation appears to be similar to JSON, but there are a few differences. Specifications of what type of device is done with a hex value. This value would determine what states the device is broadcasting. Moreover, the information regarding which network that a device belongs to is encoded in the same manner. Without the documentation of this, the intercepted messages would not be as useful to a forensic investigator. If no useful data can be obtained through the intermediate devices, then the last option would be to obtain it from the cloud.

Cloud services as the final resting place for sensor data. Depending on the policy set forward by the provide, the data could be stored indefinitely or for a short duration. However, due to issues with jurisdictions to which the provider is located at. This could result in the forensics investigator being barred from obtaining the required evidence, because foreign laws might protect providers against having to comply with seizure warrants. However, if the cloud service offers a platform for monitoring it is possible to obtain some information, granted that law enforcement can obtain the proper credentials. Smart device APIs that connects to a cloud can be accessed remotely, with software or tools. Depending on how the API is secured by its provider, it could mean that access is only granted to those who can provide the appropriate authentication. For instance, the Things Network is a commonly used API that supports gateway devices, such as LoRA gateways from Mikrotik. Their method of allowing API access, requires that the user provides three pieces of information: the application key, application id and device id. Without these credentials, no access i given. However, with the proper access, the data can be access, as well as logs to previous data. Provided that access is given to the cloud service, the next phase would be to obtain information. [Mef+17] provides a comprehensive list of manufacturers and also lists which of the devices made by these manufacturers are obtained by their proposed data acquisition framework. Their proposed framework, however, does not exclusively gather from just cloud sources, but also from the devices directly or from the controllers. Moreover, the article further states that the main method of accessing cloud data is done so through a supplied Application Programming Interface (API). Among the results

listed in the tables, only a few selected candidates were successfully accessed through the by its cloud interface. If the goal of the investigation is to obtain information off of a device that was generated in the past, thecloud would be best candidate to examine. Since, nether the sensor, nor the intermediate device keeps data outside of their volatile memory, the only area to which persistence storage would be the cloud. However, just obtaining the evidence is not sufficient in forensics. Integrity of the evidence is equally as crucial as acquisition, and IoT devices poses challenges here as well.

4.2.2 Evidence integrity

Evidence integrity is crucial for Forensic personnel to reach the appropriate conclusions about their investigation. Integrity of evidence entails that the contents of the evidence is not altered. Circumvention of integrity in this instance, refers to actions of intentional or unintentional obstruction, which leads to the evidence becoming ineligible for use by forensic investigators. There are three main categories of entities that pose a risk to the nature of evidence. The first group is the forensic investigators themselves. Methods and practices used to procure and analyze evidence is not suited for all instances of devices. When an established forensic method is used for IoT, there is no guarantee that the methods are forensically sound and the risk of data becoming useless. For instance, given a scenario where a suspect is implicated as a suspect due to their cellphone's location data indicating their presence at the crime scene, however, whereas the CCTV footage suggests that the suspect was never in contact with the scene of the crime, but rather in proximity. This particular case would suggest the potential for additional evidence, discovered in the IoT domain, to be used as a means of altering the likelihood of a hypothesis. In this case, it is used to eliminate a proven hypothesis, due to inaccurate location data. A major problem that is related to evidence is the risk to forensic examiner's and forensic analyst's mental health, when exposed to harmful material.

While discussing occupational hazards in the field of Law enforcement, forensic investigators can also experience trauma at the workplace. In stark contrast to how police officers experience issues as a result of lacking intelligence, the problems with forensics is rather attributed to exposure to unsettling material. Evidence in the form of visual and textual media could contain material, that is considered to be traumatizing to those who are exposed to it. A major category of mentally scarring material would be child abuse material. [Mul11] stated that, by its harmful nature, the prolonged exposure of Child abuse material would be an contributing factor to mental trauma for a forensic investigator. It is further inferred in [Cra+14] that the levels of secondary traumatic stress (STS) in forensic workers, who are exposed to child exploitation material (CEM) were high. In [Teh16] the same study, yielded a lower score in STS when exposed to CEM. However, it was hypothesized that the departments who were interviewed conducted thorough screening of their candidates before employment. Another study in [Sei18] on coping methods for prolonged exposure to CEM, showed that a significant portion of forensic staff seek counselling or use coping mechanisms to forget what they have witnessed on duty. In order to reduce the level of mental stress caused to investigators and forensic examiners,

automation and machine learning must be utilized to alleviate the need for human involvement. [Dal+18] proposed a machine learning method of flagging CEM. The research proposed in this article could be a paradigm change in how CEM is examine in forensics, as the machine learning would make the necessity of a human examiner less prominent. Their results, however, shows that machine learning is not performing as well as its human counterpart, and conclude that further revisions to the classifier is needed. Even if coping with exposure to such material is a probable option for the staff or counselling, the fact remains that some individuals does not have capacity to carry on. Alternative measures must be considered. An ideal solution would be to allow for screening of visual data from smart devices with machine learning to reduce the amount of exposure to CEM.

4.3 Discussion

The prior research conducted into what a law enforcement personnel require of technology, in order to fulfill their required duties, are majorly rooted in such technology's ability to serve information. The main distinguishing characteristics that sets forensics apart from field personnel is on how the data is used and when. For a field operator, the availability of such information is required to be present as soon as they are recorded by a sensor. On the other hand, a forensic analyst does not require information to be instantaneously, during a crime's timeline, to appear on their screen such that they carry on with the duties. Their main concerns is to piece together the clues to form a hypothesis as to the events that transpired before, during and after the incident. Equivalently, first responders cannot obtain sensor alerts after the crime takes place, because then they would not be able to intervene in the incident. However, timeliness of reporting information to personnel only do apply in incidents where the ability to prevent the crime is necessary. On the other hand, the necessity to provide appropriately timed reports of inappropriate parking is considered less of a priority, as this does not pose a risk to human lives or the integrity of one's property.

When it comes to work-related hazards that occurs in law enforcement, the main difference between fieldwork and forensics is on the types of hazards that these two groups are exposed to. Due to the reactive nature of field personnel, the requirements for instantaneous feedback about the incident becomes crucial. For without the proper stream of intelligence, the validity of information rating to the situation becomes unclear to the officers. Ambiguity of intelligence, cause uncertainty and doubt, and this could lead to the responders either adapting their strategy based on this intelligence or improvising a new strategy based on circumstantial opportunities. In either case, a miscalculated decision could lead to harm done to the operators or to the citizens. Moreover, these outcomes also risk putting stress on the operators, as well as the probability of causing chronic psychological scarring. On the other hand, the forensics team are less in the field during the time to which the crime is taking place, but rather is called in after the incident to piece together crime. Information obtained from a crime scene, could involve data to which the nature of the contents is deemed unsettling to an average individual,

and therefore it could pose a risk to the forensic analyst who are to examine the contents of the evidence. Another important point that isn't explicitly mentioned in the previous sub-chapters but does pose a risk to the transition over to a "*Police force 2.0*" is related to the willingness of the law enforcement to adopt a new way of practicing polices work.

Most importantly, adopting a model which utilize Information and Communications Technology. An unchanged and conservative policing culture is rooted in its own believes that personnel are more inclined to keep their learnt routines for the sake of simplicity, as opposed to learning to use technology to improve upon their ways of policing. In [TS19] it was noted that the assumption of a consistent, unchanged and generally a conservative police culture, is further from reality than was initially thought. In their study, it was noted that the Netherlands is a prime example of a police force, which does not adhere to the same culture, as depicted by British or American Law Enforcement. Rather, the Dutch police was more receptive to change. Given that the officers were more open to change and innovations, under the pretense that it would lead to a better outcome for their career.

5 Devices

5.1 A city of devices

It has become more important than ever, to integrate the fundamental processes in a urban environment with contemporary technological solutions. In pursuit of integrating information technology into a city landscape, a city aims to draw the benefits that the modern tech-solutions has to offer. One major advantage of introducing technology into a society would be for its sheer convenience. By introducing a tool as a means of completing a task, the overall difficulty or strain involved with such a task is reduced. Smart cities pose such as convenience in a multitude of activities, including automation of payment processes, which were introduced in china in 2019. In this instance, the payment is proceeded through a facial recognition system, that are facilitated by China's recent roll out of a comprehensive camera surveillance system.

Smartphones are another means to which users can utilize smart technology to commute from point a to point b. Using a smartphones location tracking capability, mobile applications such as Uber, Lyft, Juno, ReachNow and Via can connect a passenger with a chauffeur. Furthermore, transportation services and payments can be done through Near-Field Communication (NFC) on a smartphone, thus making paying for a train ticket more convenient. In homes, appliances can perform automated tasks, which could make a morning routine for a person more feasible. For instance, a smart home can detect when a user wakes up and then the hub could initiate a routine, such as turning on the coffee machine, adjusting the thermostat or citing news and weather. In industry and agriculture, smart sensors also can aid in making different processes involved in these sectors more manageable. Water levels in the soil at a corn farm can for instance be monitored with sensors, and alert when the soil becomes dry. Industry could use sensors in warehouses to communicate with robots, such as the automation that was implemented in Amazon warehouses [Dig19] All of this is made possible through the use of sensors, which requires input from the user and based on that input would perform an appropriate task. There are several areas of a city in which smart devices can be applied. In order to determine what devices can be used in police work and forensic investigations, a division of the different areas of the city must be divided.

5.1.1 Small differences, sways applications

Most cities were built with an established ruleset in mind[MK17], which facilitates how its areas are arranged, connected and divided, how buildings are structured and placed within each zone and so on. However, the buildings and facility that are built within each of the cities do differ to a reasonable extent. For instance, cities built in inn-lands on arid fields, would have a higher need to build farms to exploit its local resources. Cities built near rivers would likely

build mills and bridges. Meanwhile coastal cities would have harbors and docks for trade and export. What is essential for a city is what makes a city, but some cities have specific needs that must be satisfied, therefore the application of IoT technology would differ as well. As point of reference in this article, no city is the same. Cities do have a plethora of elements in common, which is to be believed to emerge within every city, due to its necessity by its inhabitants. These elements are expected to be present within every city as a means to serve a demand by its residents or commuters. Especially, in urban areas, where the number of inhabitants reaches a significant number, it is not uncommon for these elements to be a given; a recreational, financial, educational or otherwise required facility that a reasonable portion of those living in a city is dependent on using to a varying degree of frequency. For instance, most cities do possess some means of providing medical services to its inhabitants, whereas the extent to which these services are covered, and capacity may wary. On the other hand, cities also strive to provide the appropriate level of education, where this also may wary between cities. This may be attributed to the factors, such as population, public/private funding, the location to which the city were built. These are some of the reasons as to why a city can provide its citizens with multiple variations of the same services, whereas some cannot.

Funding, again, can be limited due to the low number of residents, which is the reason as to why it is less likely to build a large hospital in the middle of a small rural town in Nebraska, as opposed to building one in the heart of a town in the state of Maryland. It is by that fact, that the roll-out and planning of a smart city must differ between each city, and those who wish to plan out such a feat must be able to see the minor differences in facilities and needs. Only then can a city be properly equipped with state-of-the-art Information Technology installations that would best fit the needs of the citizens who lives there. What should be taken from this section is that it may be tempting to treat all cities as the same, based on the core facilities that are found in almost every cities, per definition of such. However, even those similarities may be different, to the point where servicing those facilities with smart technology could be considered unnecessary.

5.1.2 Elements of a city

To provide a complete list of all the elements that is to be expected in a city would be too exhaustive for this project. What this section aims to achieve is to set the exception of the reader as to what is most likely encountered and what is occasionally seen within the confines of a city. As a starting point the most common element of a city i the inclusion of a transportation system. Methods and means of transportation would vary from place-to-place, however, it becomes quintessential for a city to have some sort of transportation in place to be able to get citizens from point A to point B. Data that could surface when introducing smart technology in transportation services, solely relies of the means of transportation. Moreover, the extent to which data can

5.2 Surveillance

A visual overview of the crime scene can provide an operator or a forensic investigator with crucial details, which could aid them in performing their tasks. A popular candidate in this category would be the **CCTV** camera. IoT technology does not leave behind the technology of the past. In light of the push towards developing smart devices for use in cities, has reinvigorated discussions about the possibility of allowing surveillance cameras and surveillance systems of having the capabilities of more thoroughly observing and analyzing the footage to which they are observing. Cameras for use in surveillance, were previously only anticipated to be used as a means of detecting presence of trespassers, or to be used as a means of proving guilt when applicable. For instance, traffic cameras do to an eloquent degree, aid in a courtroom to prove that a perpetrator did or did not violate the traffic laws. the use of camera footage in conjunction with license plates and other information allows for this to be an effective tool of incrimination or vindication. Another example of this being the case can be seen in convenience stores and shops, where surveillance is used to catch possible shoplifters and robbers. The main purpose of surveillance cameras being that they can be utilized as a means of proving that something unlawful took place at the scene. It becomes a necessity to serve justice, especially when surveillance makes it challenging for the defendant to prove their innocence when they have been caught on camera.

However, modern communication technology, coupled with computer vision and machine learning, has allowed cameras to process more information as they are observing. Technologies that allows cameras to determine what entity lies within their field-of-view, has allowed for detection of human faces, objects and animals. In [Che+19] it was shown that the use of machine learning techniques, a CCTV camera were capable of differentiating between different kinds of wildlife using Convolutional Neural Networks (CNN). Another example of entity awareness in CCTV systems, were demonstrated in [MB19], where they demonstrated a way for cameras to search for faces, using cameras on a public transport system. In essence, the ability to use new technology to re-purpose the old, allows for new ways of utilizing existing technology for purposes, to which they were not intended to be used for.

In china, the idea of using surveillance to enforce public order has been in the spotlight in recent years, due to the sovereign state's introduction of a social credit system. This system entails that the Chinese citizens are given a score, based on their day-to-day behavior, where inappropriate behavior is penalized, while good behavior is rewarded. With this system in place, the idea is that civil order is enforced through the idea of impeding social pressure upon the citizens, with the goal of pressuring them to become upstanding citizens [OR19] [Wei+14]. By social pressure, the norm is set that persons who surround themselves with others that have lower scores, will also be penalized with a lower score, for choosing to associate with citizens of a "*less desirable character*". One of the main topics raised when discussing the social credit system and the surveillance system in China is on the extensive roll-out of smart CCTV systems. What sets these systems apart is that they are interconnected with capabilities of identifying

a person with facial detection. This allows the Communist Party of China (CPC) is able to keep records of citizens whereabouts at all times [OR19]. **Presence (proximity) sensors** is used to detect the presence of movement within a confined area. There are a multitude of applications, to which appliances rely on the detection of human presence or movement, in order to complete its sequence of actions. For instance, the use of burglar alarm systems relies on the ability for a sensor to provide the alarm with information about the presence of individuals within a home, when the alarm is activated. Another application where sensor data is required is to automatically perform a task whenever human presence is detected. This can be seen, typically in cyber-physical systems, where escalators, sliding doors and entrance bells are used in conjunction with a proximity sensors to activate a sequence of responses based on detection.

5.3 Residential Devices

Home door locks that can detect whether a person does require to enter or exit a building has become more popular in homes. For reasons, such as security, given that some smart door systems have authentication capabilities, that only allow passage when an authenticated subject is present. Another aspect of security with smart door systems is the fact that it can allow for a more secure way of checking to see who is at the door. This would involve either the use of an intercom system or an interface connected to a camera that is placed on the other side, facing the entry to show who is at the door. And convenience, since a smart door system also do possess a mechanism which automates the locking and unlocking of a door.

There are potential for investigating cases in smart homes, where information that is gathered by a smart door sensor can be used to investigate further into an anomalous incident. In [Bab+18] it was proposed in their smart home test-bed configuration, that the data that is being obtained through a smart door system, can be used to investigate the events that led up to an incident. In their example, the state of the door lock during the fire incident is checked for whether the door was locked/unlocked during the fire incident, or whether it was opened during the event. In [SC19], the potential for use of smart door sensors in forensic investigations can be justified due to the timeline capabilities of door locks. In their research, they inferred the significance of time logging in door system events and what potential it could have for the forensic investigator in a timeline analysis. For instance, the use of timestamps would help the investigator in establishing whether the door was opened and when it was opened during a criminal event.

5.4 Wearable devices

Given that most forms of communication take place on a handheld device, **Smart phones** is a relevant subject for investigation, due to its relevancy in every person's life. In same conjunction to smart phones, **Smart watches** holds similar data to what a smartphone does. They have also become a crucial platform for means of communication, either with apps that are running on a smartphone or through built in apps. **Health monitoring** either from a **smart watch** or

from a customized device, can monitor the health status of an individual who wears the device. Electronic cardiac regulators, such as the **Pacemaker** are another example of a wearable device in which useful information is stored. Pacemakers are used as a medical tool to aid patients with uneven heart rhythms; however, it has been shown that these can be circumvented. According to [FDA17], certain models of pacemakers have been proven to be vulnerable to attacks, thus allowing for a cybercriminal to tamper with the device, which can lead to death of the patient. Another example of a self-regulating medical device is the smart insulin pump. It is used to measure the patient's glucose levels and administer insulin to the automatically. Unfortunately, as with the pacemaker, this device is also shown to have weaknesses. [Bea16] disclosed that there existed three vulnerabilities on the Animas OneTouch pump system. These three vulnerabilities allowed for eavesdropping, replay attacks and circumvention of the pairing process between the pump and its remote control.

5.5 Environmental Sensors

Temperature sensors are used in a wide range of applications, stretching as far as the use in industrial solutions, to agriculture and domestic settings. One of the main attributes as to why temperature sensors is used across various environments and industries, is that it measures an important environmental value. For instance, in use in agriculture, more specific in a hydroponic growing facility, the use of thermal sensors are used to measure the levels of heat in relation to the room. With these measurements, the farmers are able to keep control of the exposed heat to their plants and can readjust water levels when required. Another example to which temperature is used is when indoor thermostats are to regulate and maintain their configured temperature settings. Lastly, a temperature measurement can be utilized in a fire suppression/alert system to which the sensors are used to signify the probability of there being a fire present in the building. In similarity to a temperature sensor, a **humidity sensing** device can monitor for the density of water particulates in the air and is often used by weather stations to detect outdoor air humidity. Another measuring sensor for air quality is the various forms of **air quality** sensors. Depending on what type of gasses the sensors is capable of detecting, an air quality sensor can be used in applications, such as detecting presence of CO2 in the air when there is a fire, and the presence of other hazardous gasses. The application of these sensors can be applied to detecting leaks, possibility of illegal drug laboratories, help in determine a fire present and so on.

5.6 City infrastructure

On the transportation side of a smart city, there are a few strides in recent years, made possible with the use of smart devices.

Location tracking and near field communication are two particular technologies that has impacted the **transportation sector** in the last few years. Given that being able to track the location of one's transport has proven to have impacted the way one use transportation serviced

today. Previously mentioned companies, such as Uber, Lyft, ReachNow, Via and Juno are amongst some of the startup companies who leverages location tracking and smartphones to connect passengers and drivers. In public transportation, trains and the metro are gradually integrating elements of smart solutions into their business structure. For instance, the state of California implemented a NFC ticketing system, to reduce paper waste generated by commuters. Smart traffic management systems using computer vision and artificial intelligence, have been proposed in [Ana+19] where cameras could be used to detect wheel-chair users and provide them with more time to get across a road crossing.

5.7 Summary

A smart city can contain a plethora of different kinds of device. What types of devices a city has depends on what the city considers to be important. As discussed in this chapter, the types of devices that can be seen in a city can range from a strongly interconnected one, or a city where smart technology has little to no influence. Regardless of how a city is built or the extent of embrasure that one has of smart technology, the coverage of a city that smart technology has is comprehensive. All listed devices shown in this chapter is by no means a complete list of all types of devices that do exist today. However, the purpose was neither to provide one to the reader, rather to illustrate some conceptual devices that do exist and how they can be leveraged in law enforcement. Furthermore, the extent to which these devices do provide law enforcement with evidence is neither a complete list. The nature of information and how it can be procured by law enforcement and used to form a hypothesis, solely depends on the creativity of the examiner and the prerequisite evidence that are present at the time. Information on a later stage can be used as evidence in different cases, where they apply. It is not a guarantee that the same data source can be used when investigating different cases.

6 Data problem and proposed application

When assessing any form of tool which is to be utilized, with the underlying goal of providing a group with accurate data. This data will presumably be used further by the group to derive the needed information to make a decision and form a strategy to execute their plan. In order for a program to achieve this, the question that should be asked is what is the program expected to communicate from what it observes? This question can further be divided into two sub-problems, namely the value of the data and its guarantee of being of value. In other words, the questions are:

1. What information is relevant to know in crime fighting?
2. What is the guarantee that the information is accurate?

Why does these questions need to be asked, let alone answered? Law enforcement are hinging on the information they get is the right information and that this information is correct.

6.1 What data should be considered relevant?

From the previous chapter, it is clear that the use of smart technology has spawned new use cases for things, that were seemingly not originally intended to communicate its own information forward. As a result of the move towards a smart world, the entropy of the data that is being generated will increase to drastic proportions. When the diversity of what data can be encountered from the emerging world of smart devices, it becomes crucial to take the data that still remains persistent across all devices. *Persistent data*, is data that are predisposed to serve a purpose that is crucial to establishing communication between endpoints, to prove ones identity for authentication, to differentiate data entries in databases or for syntactical and semantic purposes such as data-type specification. For example, in IP/TCP an essential piece of information is IP addresses, which are logical addresses to which communication devices rely upon to steer messages back and forth between endpoints. For without this kind of information, it will become hard to delineate a unique identity to an entity associated with Information Technology. Persistent attributes linked to data or an entity must pertain to a specific characteristic which is inherently common amongst all entities of similar nature, such that their identities can be distinguished.

6.1.1 What data should be treated equally for all devices?

To be able to determine what device is associated with which owner, or to which physical location is a device situated in, there has to be a data attribute to which such association can be inferred.

Even when unlimited access is granted to all devices in a smart city, or if the acquisition is done effortlessly by the police, without this key information that can put a device at a certain location, point to a specific internal cluster of associated entities (i.e. In the same house, in the same building, monitoring the same corn field, etc), the data will have no value to law enforcement.

Name	Attr. Type	Purpose	Description
Coordinate	Location	Device localization	A n-tuple of position values
IP-address	Location	Device (network) localization	Logical address of a device
Manufacturer	Identity	Device recognition	Manufacturer identity
Device Model	Identity	Device recognition	Model of a device
Id number	Identity	Owner/Device identification	Arbitrary value of an entity
Key	Identity	Owner/Device identification	Keys attributed to devices
Username	Identity	Owner identification	Identity of an owner

What good will it do to have a data cache from thousands, tens of thousands, hundreds of thousands or even millions of devices, if there is no inherent way to associate and group some devices, while keeping other devices of unrelated characters apart? Once the number of devices grow, law enforcement's ability to distinguish devices apart diminishes. At a certain point, the data just becomes a meaningless indicator that something is wrong. An indicator with no specific clarification of the problem in relation to who, where, how and why.

Specifying a location could in this instance tackle the problem and its linkage to where, and perhaps also with who the problem is issued. Global Positioning System (GPS) coordinates would be a likely candidate. However, in cities with high population density, the requirements set on accuracy is more significant than a suburban neighborhood. High density living poses issues to the leverage GPS technology has in tracking incidents. For instance, a multiple story building introduces requirements to specify the altitude on top of the requirements of providing the latitude and the longitude. The difference is that latitude and longitude would only tell the location in a cardinal direction, but it will not differentiate between apartment 3 on the first floor, nor the apartment(s) below and above it.

Another approach is to discuss the value of the Internet Protocol (IP) addresses. It was initially thought that these addresses could be used as a means to link a connection and a location to a particular entity. However, with the limitations of the IPv4 address space, that only would allow for a total of 4.3 billion addresses, the vacancies of addresses were depleted in short amount of time. Therefore, Network Address Translation was proposed as a solution to maintain a high number of address vacancies available. With Network Address Translation, the idea of remapping addresses internally on a network was possible, however, it was no longer possible to attribute an address to a particular host machine. In other words, NAT made it harder to track individual users, by limiting the unique addresses to only routers. As the number of routers grow, the need to remap the addresses for the routers became a necessity

as well. Now it is up to an ISP to assign private addresses to its clients before letting packets through. As a result, the availability of tracking IP addresses has been reduced even further. Other characteristics that could help in identifying the origin of a device would be any form of information that relates to a person or an entity.

When a device connects to a service or another device, there are some characteristics that are shared in between the two parties, which are used by both to identify themselves and the other. In Samsung's smart home platform SmartThings, an exchange of information takes place before data can be transferred to ensure that applications know what type of device is attempting to connect. A Zigbee join message comprises of information that could tell what type of device it is and what data that it intends to transfer. One particular attribute that could be derived from such a message would be information pertaining to a manufacturer or a device model. Information such as these could be used to identify what device is trying to communicate, which could be used to create a fingerprint for a format and also be used as an identifier. Since Samsung's smart home platform supports a plethora of devices it would mean that a large footprint of devices becomes available for formatting, granted that access to this is achieved. On the other hand, public data sets do also contain information that can be used as identifiers. Other data that could be useful is that which can identify the proprietor of the device. Identifiable data can be obtained when manipulating queries for crowd sourced IoT devices, on the open IoT platform SmartSpeak. The service SmartSpeak offers public directory of IoT devices that are connected to their services. All data present on this public directory includes devices, whose users have agreed to publish the information publicly. Information that is provided by their API includes identity of each channel and the identity of the owner. This is information that could be of use, when attempting to connect a particular device to an individual, although achieving so requires some additional scraping on top of a regular API query.

6.1.2 How is exclusive data obtained from devices?

Depending on what each of the devices record, the methods to which data is collected and interpreted by a program will vary in some fashion. For instance, it would be reasonable to conclude that there is a need to differentiate between a device that records the state of a door (i.e. locked or opened) and one that sets the color values and brightness of a light bulb. The need for differentiating between the two remains a problem when there is little information present to properly differentiating devices based on model and type. In the previous section, the identity of the device would be a useful piece of information which would make integration with a format more effortless, but without that information, the program that is to be developed must be able to adapt to unknown entities in some other manner.

A previous proposed solution, such as how Samsung implements cross compatibility for its devices is to require that information about the type of device and what it information it will send to be declared in a join message. With the access to these join-messages, it will become relatively easy for a program to determine what type the device is and what it is going to send

over the network. A solution which makes obtaining data from it simpler, only requiring that its metadata is known before acquisition.

With absence of a clear definition of what the data represents, the developers and users are left to deciding on their own, what the meaning of the data is and from where they originated. As there are no predefined semantics linked to the incoming data, this becomes a challenging problem. The previous section discussed the problem with authentication, and with no means to authenticate a device, then it becomes harder to identify its data as well. The issues with exclusive and common data that exist online is something that needs to be taken into consideration when developing a unified format for them. Especially when there is lack of information that could be used to make data distinguishable and useful.

6.1.3 The program

The demonstration program that is is proposed in this thesis is developed using Google's Golang (Go). The justification as to why Go suits this language is that it possesses some built-in functions which makes it easy to write programs that can handle heavy computing tasks. Go's implementation of concurrency, named go routines, allows functions to be executed as concurrent routines alongside the main program. Where these routines act independently of the main program that executed it. Therefore, it is relatively simple to write a Go program that is primed for production implementation. For a program that has to be ready to handle multiple queries from a large number of devices, a programming language that makes it simple to test and implement parallelism would be the best contender.

Main program: A synopsis

The main program that can be found at [Sel20] and comprises of modules that allows for the creating, purging, managing and debugging of connections with smart devices. However, since cloud service providers for IoT devices keep data away from public, this program is restricted to only a few sources of public and private IoT data. [1] In figure 1, there are 7 steps involved in obtaining information from a cloud instance with this program. First step requires the input from the user, such as the targeted device, service and data formatting. Next in phase two the program collects this data and analyzes it. Step three is when the program has determined the target of the acquisition and sends a request to the cloud server. The cloud server in phase 4 returns the appropriate data. The fifth step is when formatting of the raw data is conducted by the main program. JSON files with configured formatting rules and targets are used in the program to sanitize the data after user specification. Step six performs additional actions on the data. These actions comprise of a set of smaller operations that can be conducted with the data, based on user preferences. After the appropriate actions are taken on the data, the data is outputted to JSON and stored in step seven.

[1] https://github.com/matsse/SmartPolice-Interface/

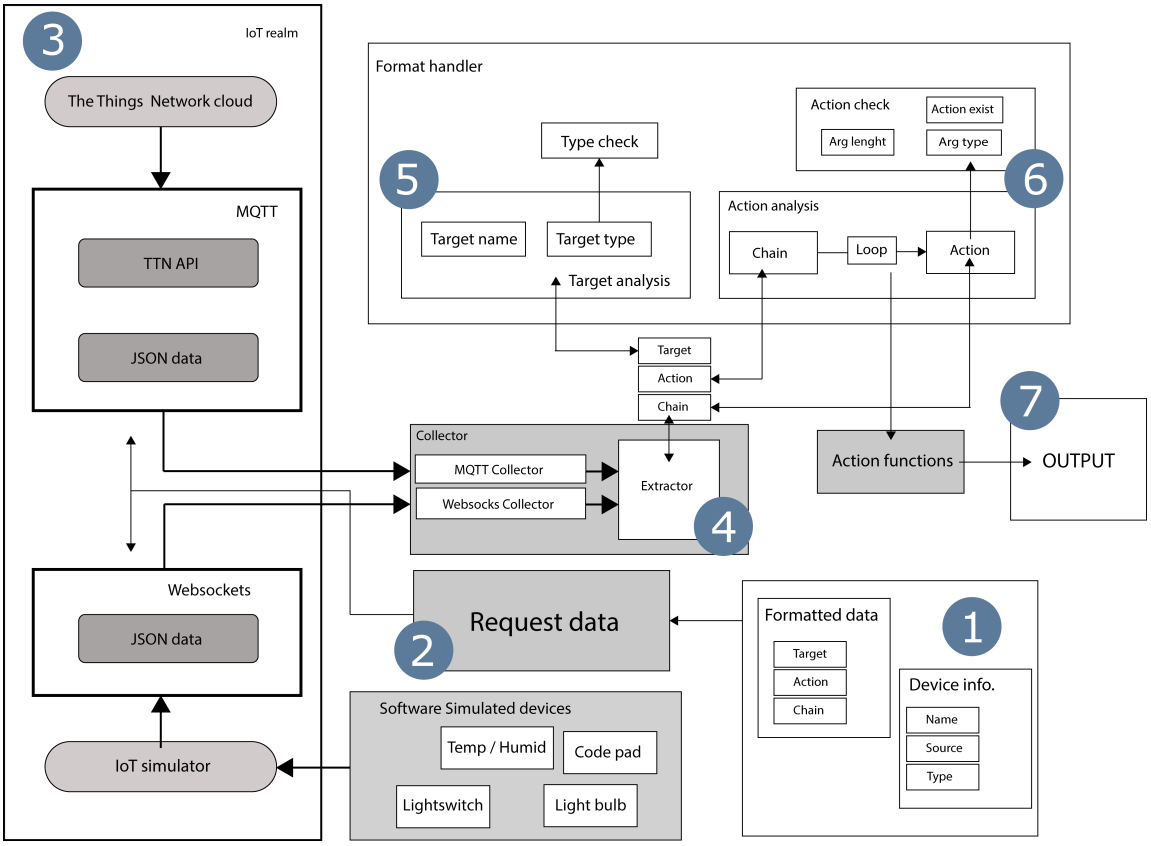


Figure 1: Main program explained

Main program: First execution

When first executed, the program will be empty with no devices connected to it. It is programmed to add devices based on the service provider that the user specifies. For instance, the Things Network is the main online service which is used in the program. As such, the specification is intended to initialize the correct information that is sent to the provider, such that access is granted to the cloud. Once this information has been entered and a connection has been established the program will save a file with that information and re-enter it into the program upon next startup. The file that is generated will contain the information about the service provider, the specified format, last data entries from the device and other important information. This makes it simpler, as one does not have to re-issue the same information upon execution of the application.

Main program: Format specification

Once the device is added to the local program instance, the user can then specify a format for how the program is to interpret the data that the newly created device will upload. The general idea of how the formatting of the information is done, works in the following way. An output that is generated by TTN is in a JavaScript Object Notation serialization. As shown in 1 the structure of a JSON file comprise of a key and value pair, where the key is used to index for a value. For instance, a program asked to lookup the "*firstName*" key, will return the value of "*John*". Equivalently, a query for "*lastName*" will return "*Smith*" and so on.

```
1 {
2     "firstName": "John",
3     "lastName" : "Smith",
4     "age" : 25,
5     "info" : {
6         // [Omitted info]
7     }
8 }
```

Listing 1: JSON example

This makes it simple to index and interpret information stored in JSON serialization. Hence, its popularity alongside other serializations, such as Extensible Markup Language (XML) and Protocol Buffer (Protobuf). In order to obtain any data from a smart device who uses JSON, the approach is to issue a query for its keys to get the data.

The proposed method of obtaining information is to define the keys in a data format that a Go program can read and interpret. The idea is that a pre-defined list of instructions will acts

as a recipe for how this computer program should assess and obtain data. The best option to providing a target for the program is to define a keyword for the program to use as an indicator that a target entry is required. For simplicity, the keyword used to issue a key/value query is denoted with the phrase "*key*", followed by a delimiter and the actual target value.

```
"GetName" : "key:firstName/"
```

Listing 2: Format example

With the use of keywords in the format means that the program can determine what course of action to take on the data based on what keywords are supplied by the format. Keywords makes handling some issues more manageable, for instance when a user wish to specify the location of the entry in a larger JSON document.

```
"info" : {
  "skills" : "Good Listener",
  "languages" :  ["C++", "Go", "Rust", "Python"],
  "education" : {
        "primary": "Bryant elementary",
        "secondary": "Oleander High School",
        "higher education" : {
            "Undergrad" : "Crestmore College",
            "postgrad" : "Hills Institute of Technology",
        }
    }
}
```

Listing 3: Info section from 1

One of the issues that would arise is the fact that some entries are enclosed in other structures, thus forcing the lookup to recursively go over all entries and child entries until a match is found. In a real world scenario, this approach to searching for an entry is not ideal, especially if there are a plethora of the same devices present.

To combat this issue, the proposed additional query method is titled "*path*" and replaces the "*key*" method.

```
"GetCollege" : "path:info.education.higher education/"
```

Listing 4: Format example

In 5 the syntax for the patch is denoted with a period (.) as a delimiter, to ensure that each of the keywords are distinguishable to the program. As also seen in 2 and 5 there are three characters in used in the value field. Periods, colons and forward slashes are intended to act as delimiters for entries. A color separates the keyword from the argument, while the period separates the variable, if it comprises of multiple values. The forward slash separates keyword/argument combinations when there are more present in the entry. Additional combinations on a single entry is useful for when there is need for more operations to be performed on a single entry, such as performing a type assertion.

```
"GetCollege": "path:info.skills/type:string/"
"GetSkills" : "path:info.skills.languages/type:[]string/"
"GetAge"    : "key:age/type:int/"
```

Listing 5: Type assertion example

Type assertions are much like a type casting in Rust, C++ and C, where a variable of a particular data type is converted into another data type. For instance, some compilers might reject performing basic arithmetic operations on two variables, if they are not of the same data type. Type assertion would remove this problem, by assigning the variable the same type as the other variable. Asserting variables that have data types that of arrays and slices, the appropriate syntax requires a prefix with square brackets ([]) before the type specification. The purpose to formatting entries this way is to allow for more flexibility in how data is interpreted and handled by the application. In contrast to developing an application that requires a predetermined list of conditions to be met, which is inescapable in many instances when developing code, this program aims to let the format dictate what the program does. For instance, the use of hard coded definitions of target and variables causes restrictions as to what a program can do, based on the existing information compiled in its binary. Functions with sequences of instructions would be limited to the exact chain on instructions that i was created with when it was compiled. Conditionals might alter the flow of instructions, but it would still remain the same. When this project is intended to provide a program that would potentially be used in applications with machine learning, computer vision and high frequency networking in mind, the notion of implementing a flexible way of executing alternating chain of functions becomes

more apparent. A proposed implementation in the main program is the use of format defined actions and format defined action chains. Actions comprise of statically typed functions in the binary that can be executed with a data entry using the "*action*" keyword in the same format as the other keywords.

```
1    "GetAge"        : "key:age/type:int/"
2    "GetAge.Action" : "action:convert2float"
```

Listing 6: Action example

Actions called from a JSON entry is supposed to be a way to issue a custom macro from a program, without having to run different data through the same function. Instead of writing a function in the program itself, this solution proposes writing a set of smaller functions that can be invoked by the user. Flexibility such as deciding which Action to run on which data makes it more dynamic, in the sense that no data will necessarily have to forego the same execution routine, unless the program is instructed to. For instance, if the user wishes to compute daily bike commute statistics (int) and refrigerator temperature measuring (double/float), they will have the opportunity to do so. Even if the two data sets use different data types, the use of actions would give the opportunity to convert one data set to match the other. Moreover, another situation where there is a split population of similar devices, where some use encryption and others do not, the use of actions could provide a method to decrypt information prior to use. The problem here that would transpire is that some devices might use encryption, while some does not. Furthermore, the encryption method is also something which could differ from devices, namely the algorithm used and additionally these algorithms would require additional information to function, while others do not require such information.

```
1  var Action map[string]  func(..., varN type) (..., ReturnType)
```

Listing 7: Typed map with function type and

This is similar to how JSON functions in the sense that the association in a map container requires a key value pair. The thought was to create maps containing a criterion and then assign the value a function type. Listing 7 shows how this implementation is declared in a go file. A map declaration requires that the key type and value type is declared. As will soon be shown, this implementation does have some restrictions.

```
"int8" : func(in interface{}, typ string) (int, error) {
        return int(int8(in.(float64))), nil
},
```

Listing 8: Actions declared with a string as a key

One of the inconveniences of using this is that the function must be re-written for each map entry or for each condition that must be met. Listing 8 shows a snipped of a type assertion function that can convert all number types into integer-type. A function that is required, because of Go's limitation on automating number conversions from JSON to primitives. All numbers that are collected from a JSON document and placed into a generic interface (non-type) will automatically be casted to float64, regardless of their actual data type. Given that Go is a statically typed language, it means that variables and their data types are defined at compile time. After compilation, the type will remain the same for a given variable, meaning that one is not able to change it or use it in functions that requires a different data type as input or output. In order to address this issue, the Go developers implemented the *Interface* type to make it possible to define variables with no data type explicitly declared at compile time. As seen in Listing 8 the use of this type is present. Moreover, the use of interfaces is a ubiquitous occurrence throughout the program. Interfaces are an important decision to make, as it allows for collecting any form of entries from a file with no type limitation. For instance, if there are exist devices that records the same type of environmental values, but store its records in different manners, it will be difficult to marshal (read file into program) this data into a hard-corded data structure with explicitly defined variable/types. In essence, when there is a need to not restrict the type of data that can be handled, the use of generic non-types is the best option, although it do have some limitations. One of the most notably limitations, which has been discussed previously is that a compiled program is no longer able to differentiate the implied type of an interface. Meaning that functions that requires one type will refuse to allow interfaces, event though that interface is inherently of the same type. For instance, the use of maps helps to assert a type to the data that is being processed. When there is a need to perform computations with the collected data, then the limitations on types must be handled. If more computations on a single data entry is required, then this becomes significantly more challenging to implement. The last proposed implementation in this project is the use of chains, which are actions that are executed in the order they are specified to. When a chain of actions is required to be executed on a data field from a JSON file, the user must declare that in the format specifier file.

```
"GetAge"        : "key:age/type:int/"
"GetAge.Action" : "chain:convert2float64.sum.times.save"
```

Listing 9: Chain example

Chains and its primitive components, the actions, is a relatively new idea that is in its infancy, where the idea is that data operations will not be consistent in all situations. For instance, some projects that use this application would require the calculation of decimal numbers, while others would prefer to prep data for machine learning, and then there could be need to forward data to another instance. For this very reason, that projects require different operations and functions to be executed on a set of data in a particular order, the norm should be to provide an open and flexible environment for the user. For instance, listing 9 shows a chain that obtains a piece of information and runs a chain on it. The actions in this chain is delimited with a period and is executed in order from left to right. One of the justifications for having a chain would be that some data needs to be corrected before it is used in an alerting system or for research purposes. An example of this could be data is recorded in different units of measurement. For instance, one sensor might record in Celsius, while another in Fahrenheit. Which are units measured in the Metric system and the Imperial system respectively. If there were no methods in place to adjust for these differences, then it could lead to wrong data being accumulated and stored, which could result in wrongful termination and decision making based on an error. For law enforcement and forensics, a Crime Event system that is trained in a similar manner to an IDS could potentially report false positives as a result of processing data sets that are different in nature.

```
"action:DecryptAES(self#,sref#key,sref#iv,sref#blocksize)"
```

Listing 10: Function with arguments example

Function calls that require the use of additional arguments the proposed syntax used in the development is to use a parenthesis enclosure syntax to signify which action requires additional arguments to run the function. In the example value shown in Listing 10, a singular function call is made to a decryption function with the intent to decrypt the arriving data.

6.2 How would the data have to be presented?

All of these components together form a sequence that sets out to done thing: create a new output. Since JSON was the primary tested subject and the most familiar serialization type used, the selected output of the data should remain the same upon departing the program, as it was when it arrived. Since law enforcement and forensics requires comprehensible and organized data, which is something that is seen in JSON, there is no need to make alterations to the presentation of the data. Rather keep it as is and filter out all irrelevant entries before feeding it through. Moreover, the flexible nature of this presentation format makes it easier manipulate and make changes when needed.

```
{
    "Data" : {
        "entryX" : {
            "Temperature": 34,
            "Time" : "2020-01-02T15:04:05"
        },
        "entryY" : {
            "Temperature": 31,
            "Time" : "2020-01-02T15:05:05"
        },
    }
}
```

Listing 11: Output example

For instance, if an output file has more than one recorded instance of information to store, this could be stored within the same file, requiring only to push an additional entry to the file, with specifications that can set entryX apart from entryY. Something which is achievable by separating these entries into individual contained values. Moreover, when there is need to add additional values to the output data, then an additional field can be appended to an entry. While formatted outputs, such as Comma Separated Values are more suited towards Artificial Intelligence and Statistics, their formatting is less ideal for establishing identity of the previous data beholder. Moreover, the outline of CSV allows only for the assembly of data in a table-like manner, where attributes are arranged in columns[Y05]. Use cases, such as machine learning, data mining and model training, a format such as CSV would be ideal, as it is simple to parse and assemble its data for use in processing. However, CSV files do have some limitations on use. First is the strict nature of data assembly, where the data must be arranged in a row/column like manner, similar to that of a spreadsheet. The proposal of using JSON, contrary to using CSVs,

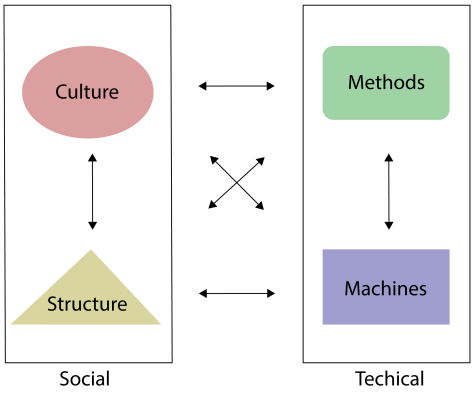
gives the user freedom to shape the data output more freely, while maintaining a reasonable structure to the overall data. Missing entries are not an issue and will not require the inclusion of a blank field, such as it is required in a Comma Separated File structure. Equally as intuitive in JSON, adding an additional field to a data entry is better in JSON, as it will not require other fields to add this data field with an empty value, for the document to be compiled in another program.

6.2.1 Summary

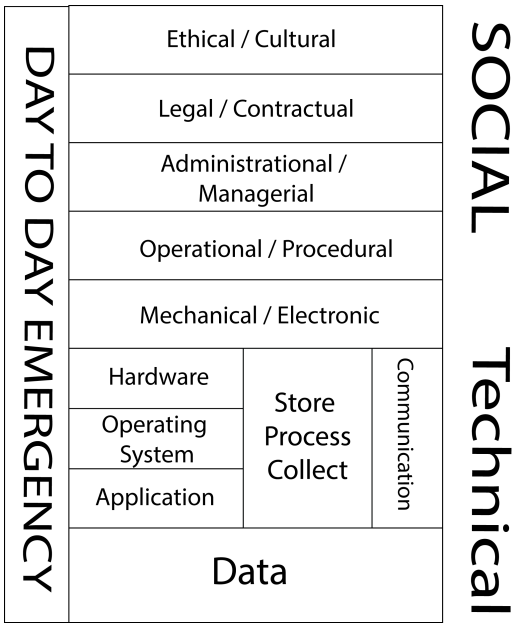
The project artefact that was proposed in this chapter, attempts to address the issues faced with assembling a format for smart devices, whose structure and contents are unknown. A major setback from using smart devices as a means to collect intelligence in a smart city, is the lack of mechanisms that can be used to fingerprint devices. When such a mechanism exists, this project would be more capable of adapting to newer devices as they are added to the database. However, for the time being, this is merely a demonstration of a program, that a user can utilize to arbitrarily define what data to collect. In essence, this program can only observe data feeds from devices to which its users have instructed it to. Automation of adding a device would require even more coding and that a proper ontology on smart device fingerprinting to be researched first. This artefact also proposes the use of actions, which are inspired by how libraries such as Go's Extensible Access Control Markup Language *GoXACML* assigns roles and conditions to user credentials. However, this program applies this concept different from XACML. Actions are not used to determine authorization, but rather to be used as a means to adjust data when there are inconsistencies is measuring systems. Additionally, when computer software requires that data is structured in a particular manner, these actions were intended to accommodate for such requirements.

7 Legal, ethical and security concerns

In a society it is important to evaluate the impacts that a proposed concept has on the society to which it is being introduced in. Before evaluating the impact of a proposed artifact, an overview of the environment is required to properly paint a bigger picture of the landscape that could be affected by the product. Since the artefact proposed in this dissertation is an implementation in software that transforms input data and generates new output from it. Additionally, the concerned groups who are affected by this artefact mainly involves law enforcement. But another group of individuals who also are impacted are the perpetrators and the victims of a crime. Though not directly impacted directly by the software or the formatting, but rather through the actions and decisions taken by Law Enforcement in light of using the framework. While there is no formal category of professions to assign to criminals and those who fall victims to a crime. Crimes know now bounds in terms of class, professions, gender or nationality; therefore the assumption is that all have to be taken into consideration.



(a) Socio-Technical model [Kow94]



(b) Security by Consensus model [Kow94]

[Kow94] introduced Security by Consensus (SBC) in 1991 as a part of the Socio-Technical framework. The Socio-Technical

framework is a model that comprises of a dynamic component and a dynamic component. The dynamic portion of this model is the Socio-Technical System (STS), which is a model to describe the interactions between human (Socio) and machines (Technical). The Socio and Technical components of the STS has each two sub-components; culture and structure in Socio, while Technical has Machines and methods. STS describes the interaction between different systems as promiscuous, where all components can interact with each other. In the static portion of this framework, resides the SBC which deliberates more in detail on the Social and Technical aspects with focus on security. The Security by Consensus model is a security model used in any organizations to analyze the overall risk. It is a holistic model, meaning that the elements in the model - as shown in figure 2b - affects each other and by extension the security of the organization as a whole. In the model, there is a separation between the Social aspects and the technological aspects of security. Each of these aspects are represented as classes, both of which have sub-classes attributed to them. Among the Socio-Technical sub-classes in the SBC model, the topics covered in the social class are: Ethical, Legal, Contractual and Operational. While areas covered in the technical class are: Data, (Store, Process and Collect), Application and Communication.

Issue	**Class**	**Sub-Class**
Data Retention	Social, Technical	Legal,SPC
Breach Notification	Social	Legal, Ethical
Right to Privacy	Social, Technical	Ethical, Legal, Data
Surveillance state	Social, Technical	Ethical, Legal, Data
SEC: Input Sanitation	Technical	Data, Application, SPC
SEC: Authentication	Technical	Data, Application, Communication
SEC: Access Control	Technical	Data, Application, Communication

Table 1: Issues and their place in the SBC model

In Table 1 the following problems covered in this chapter covers issues in the Social class and the Technical class. The first focus in this chapter is on this projects ability to comply with data protection laws.

7.0.1 Data retention

In recent years, large companies which relies on the utilization of its user's personal data has experienced a flurry of Cyber-incidents, which has led to the unintentional disclosure of data to adversaries. In 2017, the Financial Credit Audit agency Equifax had their systems circumvented by unknown actors. This incident resulted in the exposure of records pertaining to 145 million U.S customers. These records included personal information, such as credit rating, Social Security Numbers, birth dates, addresses and full names. Another incident from September in 2019 showed that Facebook also unintentionally left an unsecured server with 420 million of their user's phone numbers online. Later it was also shown that an additional unsecured server was discovered in December. This server had approximately 250 million user records. These data breaches have prompted sovereign lawmakers to establish a strict legal framework, for how user data on online services are handled by their data providers. In the European Union, the current legal framework that governs over handling and storage of data, is the General Data Protection Regulation (GDPR). This framework was put into effect in 2018 and provides a comprehensive list of requirements imposed upon data custodians, in relation to how they processes, stores and collects user data. In conjunction to data retention, meaning how data is stored and for how long it remains on a server, the GDPR lists the following requirements:

1. Data must only be stored for as long as it is needed.
2. The extent of retention must be decided and justified by the custodian.
3. Establishing polices on how long data is retained is required for compliance.
4. Renewal and anonymizing of data must be enforced at a recurrent basis.
5. Assess challenges with retention and comply with users right to erasure.
6. Extended storage is allowed if the intentions are for the sake of posterity (archiving, etc)

These points forms the basis legal framework for how the controllers of a mass surveillance system has to abide by. The core ideals of the proposed system to which the research is a part of, builds upon the principle of *storing for as long as it is needed*. In essence, this project aims to comply with the first requirement of allowing only storage of data until their intended purpose is achieved. However, the notion that were discussed in the preface to Research Question 3 about the use of data to de-anonymize other data would be a breach of the fourth point. Mainly, due to the intention of this feature, being to resolve for an identity with the use of additional data. The need to write policies on max the retention time for data fusion must be satisfied, in order to limit the use of data for identifying subjects. With the absence of a formal definition on lifespan of the data, this could be seen as a contradiction to the compliance with the first point as well as the third point. The custodian is required, according to the second point to explain the justification of the extent of lifespan. This must be done with policies established and must clearly define which data can be retained and for how long. Moreover, when the

data is to be used to train machine learning models and used in the *crime reporting system (IDS)*, the required retention time could wary. This would mean that the policies needed must set a reasonable time frame for the training and Incident Detection. If there is no such policy present, the time spent in processing could exceed any arbitrarily set retention deadlines. On top of all these issues, the last remaining problem is to maintain communication of the subjects to which the data relates to. This is done with the goal of acquiring the consent of the user, before the data can be stored. An automated surveillance system, that utilizes data sources from a multitude of platforms and manufacturers, have no such capabilities to inquire consent from its subjects directly. Another important regulation imposed on organization is the responsibility to report when an incident has occurred.

7.0.2 Breach notification

Once a system has been breached, it is up to the organization to swiftly inform the authorities about the incident within a reasonably short time after the organization has discovered one. According to article 33 in the General Data Protection Regulation, the max amount of time to notify authorities is 72 hours after the incident has been detected. Furthermore, this article also states that the internal notifying between a data processor to a data controller, must take place immediately upon discovering the breach within the organization.

Aside from the deadline requirement of disclosing an incident, the nature of the information conveyed in the disclosure should contain information about the breach. First point to include in the notification is what was breached and where it took place in the organization. Second is the contact details of the protection officer(s) and the amount of records that were affected. Third point is the determination of what impact the breach will have. The last section that has to be disclosed is the steps taken by the controller within the organization to ensure that the incident is properly mitigated. A significant portion of this project will hinge on a system that could automatically process and handle data as it is fed through the endpoints. Some of the issues that has yet to be discussed in conjunction to this project is whether it is appropriate to allocate humans as a means of ensuring that incidents are properly moderated on the internal systems. Human moderation in this sense, must remain heavily restricted in terms of what they can and cannot access on the internal systems. And therefore, a proper policy must be establish, which properly defines the roles of Incident Response personnel employed to watch over these systems. As the following section will discuss, a failure to restrict and define roles of such personnel could lead to improper disclosure of data and therefore ones right to privacy would be infringed.

7.0.3 Privacy

Progress in Information Communication Technology has enabled information about individual's daily activities to become more open to the public. Data gathering through the use of social media and smartphone applications, has given data brokers and advertising firms the ability to profile its users interests by how they interact on online platforms. Smart Technology is a

catalyst that enables even more information to be used in profiling individuals. This is because of the contents that these devices are capable of recording, which could have the potential of exposing even more information about a person. By collecting IoT data from private and public sources, the data controllers run the risk of infringing on the rights of citizen to having a private life. No immediate other parties, private person nor civil servants, should have the ability to obtain knowledge of one's private life using technology as a means to gain such knowledge. In introducing an implementation of collecting and processing data from sources that broadcast data that could be considered intimate in nature, the questions regarding the risk of breaching privacy rights must be considered. First problem that could lead to an infringement of privacy rights is apparent if the system that is being implemented does not exclude the human element from the monitoring of the data streams that are being targeted by the proposed system. In the previous topic, the problem with assigning human employees to act as notifiers of potential breaches is that their access to the data must remain non-existent. Leaving only the incident feeds available to these individuals and keep the rest isolated from being disclosed to them. Another important question to ask is to what extent would an individual's rights to privacy be infringed on if the subject conducting the monitoring is not a human, but a machine? With no actual human beings monitoring the data feeds, the question as to whether one privacy is being infringed upon or not should be asked. Especially when the entity that is assigned to assess the data, is a complex computer system, comprising of Artificial Intelligence, Data processing software and Computer Vision capabilities. The distinction between a human and a computer must be deemed different, if the notion can be altered to condition citizens to understand the less severity of surveillance being handled by a machine as opposed to a human.

7.0.4 surveillance state

Surveillance in the traditional senses involves observing subject, who are presumed to be of interests due to being criminals or a threat to a nation's security. It has been practiced throughout history, ever since the first civilizations established some form of government, coupled with a culture that builds upon military might. Terry Crowdy stated in [Cro05] that in ancient times, spies were deployed behind enemy lines, with the goal obtaining information, so that their allies would gain the upper hand in a battle. However, the practice of surveillance on a large-scale was rarely a commonality before the 20th and the 21st century. Today, many forms of large scale surveillance is being conducted by government intelligence agencies around the world. Recent documents leaked shows that agencies, such as the U.S National Security Agency (NSA) and the Central Intelligence Agency (CIA) are actively developing ways to achieve domestic and foreign surveillance. As evidenced in publications by Edward Snowden, the NSA were actively peering into the lives of American and foreign nationals using ICT. Other classified documents leaked in 2017, in the series dubbed Vault 7, further sheds light on CIAs campaign to develop an arsenal of malware and exploits targeted at all forms of computer systems and across all major Operating Systems. The nature of how their information is obtained is through circumvention of systems or cleverly manufactured methods of tricking targets to get systems

infected with malware [CIA14] [CIA16]. The issue with the way surveillance is conducted by these agencies is split into two main areas. First major issue is that the way information or access is obtained, is done by gaining the system or coercing a target to circumvent their own security measures. This form of obtaining authorization of a resource is in its very nature considered predatory and, in many instances also considered illegal. The US unfortunately do have laws which exempts many legal restrictions imposed on surveillance practices. In the 4th amendment of United States of America [U.S. Const.], it clearly states that citizens are protected against "*...any unreasonable search and seizures*", which means that any unwarranted surveillance is prohibited. However, the inclusion of the U.S Patriot Act of 2001 [US 01] has loosened the requirements of issuing a warrant. A requirement that prohibits wiretapping without obtaining a warrant, is stated in the Wiretapping act of 1968 [US 68]. Moreover, in March 2020 the patriot act [US 01] were updated, adding less restrictions on warranting wiretapping in section 215 and now also allowing no warrant to obtain browsing history in section 216. What is important to note here is that the restrictions on surveillance in the us is becoming less prevalent as the focus on ensuring national security becomes more crucial. This project regurgitates the same narrative in some sense, that the security of the people could outweigh the need for privacy and to keep prying eyes away from private lives of citizens. But the difference here is that this system aims to gather information and process it only with the pure intentions of seeing whether there are possible incidents occurring. So, the issuing of warrants to pressure providers or secretly surveying citizens are not the goal of this project. Rather the goal is to observe data once and discard all that was observed, if the nature of the observation is considered benign.

7.0.5 Why software security is important and why it must be addressed

Given that this project promotes the use of new data sources for the use in fighting crime and restoring justice, it becomes a crucial objective to address the insurmountable difficulties of securing the code to prevent abuse or leakage of data. The organization, Open Web Application Security Project (OWASP) defines a list of the 25 most common weaknesses that are occurring in computer software. In the following sections, all applicable vulnerabilities that could occur in this project will be discussed. The focused vulnerability types will be linked to processing and communication of data, as this is the most prevalent components seen in this artefact.

7.0.6 Code security: Improper validation of input

Development and maintenance of the real-life incident detection system must be secured in such a way, that its core component does not facilitate the possibility of abuse or circumvention. One particular issue that is commonly seen in applications that handles data and information is the improper sanitation input.

If the application is set out to read the input from a user, or process data from files or traffic, the assembly of such input can be used to circumvent the application.

```
<form method='get' action='index.php'>
<input name="search" value="<?php echo \$_GET['search'];?>" />
<input type=submit name='getdata' value='Search' /></form>
```

Listing 12: Example of form field with improperly configured form

In Listing 12 from [Tel11] the input field is improperly configures, where the input is not filtered when the user fills out the form and sends the query to the server. There is no filtering of the input, so there is nothing stopping the user from injecting JavaScript into the form.

```
http://servername/index.php?search="><script>alert(0)</script>
```

Listing 13: Bad input example

In Listing 13 the user can issue bad search queries, such as the output URL generated when injecting JavaScript into the field. The result is that the users who will use the same feature on this website will get an alert popup in their browser. Not just the filtering of input that has to be taken into consideration, another issue that could occur is when input is not properly checked for length. Languages that checks for input of a fixed size requires to properly check that the input strings are of the appropriate size. When the logic in a function that checks the length is not properly configured or when the string exceeds the allocated length, the program will not execute accordingly. In some instances, an improper execution sequence could force the program to crash. In the Go example shown in Listing 14 demonstrates when a function is not properly configured to measure the appropriate length, according to the string that it is programmed to look for. It is an example of off-by-one, where the length does not corresponds to the targeted string.

```
1 func ParseComplex(data [] byte) bool {
2     if len(data) == 5 {
3         if data[0] == 'F'  && ... && data[5] == 'T' {
4             // checks for the string "FUZZIT"
5             return true
6         }
7     }
8     return false
9 }
```

Listing 14: Improper length check of byte arrays [1]

When the byte array of "*FUZZIT*" is processed in this example function, the program will return false, even when that was the wanted input. The reason why this problem occurs is that the conditional used on the length on line 2 in is one byte less than the length of the string. An example of out-of-bounds bug can be seen in *CVE-2017-15672*, where the FFMPEG media library is targeted [VUL17]. The input MP4 media file, when manipulated to exceed a limit set by the media library, the result will cause a memory corruption, that would allow the attacker to execute arbitrary code.

Another way in which input can be improperly configured is when a structured file reader receives a input that violates that rules set by the file formatter. Examples of formats that are prone to bad input would Extensible Markup Language, JSON, media files such as Portable Network Graphics (PNG), Graphics Interchange Format. Other inputs that falls into the same category as files are input strings that are anticipated to follow a strict formatting rule. Uniform Resource Locator (URL), File paths, Hardware addresses (MAC) and Internet Protocol addresses (IP) are some examples of formatted inputs. When any of these are corrupter or malformed the programs that are developed to read them might end up crashing. [Net18] demonstrated that libraries that are designed to read formatted input can be tested when using automated fuzzing of input values. Their experiment showed that when a Classless Inter-Domain Routing (CIDR) value in an IP address were malformed, the standard Go library would crash when reading the input.

These three concepts of improper input validation are problems that must be considered when assessing the JSON formatter that are used in this research. Input validation in reference to issuing calls to functions is one topic that would be applicable to this project. Granted that this project allows for the use of both singular calls (Actions) and calls to a sequence of actions (Chains), the thought would be that allowing for Actions to be run would be counterproductive to security. However, the program does not allow for any function to be executed. The way that functions are allowed to be treated as Actions is determined by a hard coded list, which restricts

the allowed functions to only necessary operations needed to impose on the data.

```
1   ... : "Action:os.Remove(sstring#main.go)"
2     // Attempting use the Remove method from the "os" package
3     //to make the program delete the main.go file
```

Listing 15: Action that attempts to make a syscall

In the listing 15 the command that the JSON input attempts to execute is a system command for the deletion of the main file of a Go project. Since this command is a method that can only be invoked from the "*os*" package, hence why os is listed before Remove. However, the use of periods (.) are only reserved for chains in this instance, so when the program attempts to read the Acion, no call to "*os.Remove*" will be made.

Figure 3: Error message from running input from listing 15

On the other hand, the program will throw an error and return a panic message with a trace of the call to which caused the panic. Figure 4 shows what the program returns when an unlisted function has been executed. The first part of the error message details that the program tried to read the memory address of a variable that does not exist. In the second part is a trace of the functions called, which led the panic to occur. Using the call trace, with the error message reveals that the variable that the program was accessing were the map that is being used to store information about the available functions.

```
15
16  var AvailableFunctions  map[string]ApplicationFunc =  map[string]ApplicationFunc{
17      "sumx": {
18          Name: "sumx",
19          Args: []string{"sint", "sint"},
20          Description: "Sum two integers together",
21      },
22      "send2Ip": {
23          Name: "send2Ip",
24          Args: []string{"none", "sstring"},
25          Description: "Send json data to an ip address",
26      },
```

Figure 4: The map value that caused the crash 15

While no function call was made, the program still stopped running as a result, so the current configuration do still have an issue with improper input. Because if a function is called, when the program has not allowed it in its list, the program would "crash", meaning that error handling must be implemented to properly mitigate the issue of unnecessary panics. While abusing an implemented feature in a program by calling unsanctioned functions is not possible from using actions, but other unaccounted issues would be that length and malformed JSON documents could still allow for abuse.

Discussing the abuse of using malformed JSON documents would potentially be an issue for the formatter, because the formatter expects a certain pattern when reading the file and placing it into a temporary map structure. One particular vulnerability of this program that is linked to marshalling JSON documents into maps is that when a file is corrupted, the marshalling method will raise a panic signal. By removing a syntactical operator from a document and feeding it into the program, an expected reaction would be that the program will be confused and inevitably crash. Error handling, however, could also be a solution to this problem, but simply rejecting marshalling of corrupted data with the error handle returned.

The same issue is applicable to abuse of lengths of inputs, but this issue is less prevalent in this document, as opposed to the two latter issues. One issue that could be present in the program is the functions used to determine the number of input parameters for an Action. When a function call is made through an Action specification in a JSON format file, the number of required inputs must be present in the format. When the program is to analyze the function call, it preemptively separates the argument container (parenthesis) and counts the number of parameters. Parameter lengths are used in the program to issue the appropriate commands to execute an action. Counting the number of input variables for a function is considered necessary for the action to be executed. Even before the appropriate action is searched for in program, the number of arguments must be determined. It separates the functions into groups based on this number and will execute the wrong associated function if the count is wrong.

```
type User struct {
    Email string `json:"email" validate:"required,email"`
    Name  string `json:"name" validate:"required,min=2,max=100"`
}
```

Listing 16: Validation fields added to struct entries

In order for this program to be secure against malicious input, the program must implement a mechanism that can check for appropriate data as it arrives on the host machine. A strategy that would be ideal in this scenario is to utilize validation checking of input as new data spawns. As shown in listing 16 there exists a method in Go to validate the data in a JSON file when it is marshalled into a struct. A string input can be appended towards the end of a struct field where, the requirements of the field input is declared. When a marshal function comes across the validation key it will check that field for these conditions before entering the values into that struct entry. Unfortunately, since this formatter is more focused on implementing flexibility, the use of structs were not seen as ideal and maps were therefore used instead. Thus, the alternative method of validating input would have to check each entry manually. Rather than checking for bad input, the program would only need to check for appropriate input and discard entries of they do not conform to the rules set by this program.

7.0.7 Code security: Authentication

Given that this system is intended to act as a form of Intrusion detection system for reallife events, it is pertinent that the mechanisms to which authentication of first responders, emergency personnel and forensic investigators is secured. At this moment the application in of itself does not implement any form of direct communication to other users of this application, nor does it enforce an connection to a centralized computer system. The idea of developing this system further, however, would bring up the question as to whether a connection between other entities internally should be allowed. All forms of communication in between Law Enforcement Personnel, forensics staff and between other precincts would be considered important. Information of formats, criminal cases and incidents are information that professionals should have the ability to share internally. However, with this being a need, the proper mechanisms to enforce proper authentication of users is required. With absence of an implementation, this still cannot be directly addressed. However, the types of communications conducted in this project still poses challenges to the concept of authentication.

```
"ttn": {
            "appID": "repository",
            "devID": "hue",
            "appAccesskey": "[omitted key]",
            "appEUID": "[omitted 8 byte app ID]",
            "devEUID": "[omitted 8 byte device ID]"
        },
```

Listing 17: Cloud credentials hard coded in a locally stored JSON file

Aside from internal communication within the organization, the only form of communication that have been used in this program thus far, is the communication that is necessary to acquire device data. A major issue with the communication between the application and the Things Network cloud is that this service requires authentication. In the proposed implementation of the format, the credentials that is being used to connect is stored in a JSON file in plaintext. Upon the startup phase of this application the files with credentials are loaded, such that existing devices are automatically loaded into the program before the prompt appears. The storage of credentials in this manner is a security issue according to [CWE20c], where the storage of hard coded credentials could be at risk of exposure to adversaries.

Storing credentials in a readable format or transferring this information could be a potential for leaking the information to other users, who are not to be in possession of this information. Therefore, this system have to address the possibility of changing the way credentials are handled in a future revision. It must be a priority, even though it being outside of the scope of this research.

In a hypothetical scenario, if this system were to be acting as a relay between smart devices and law enforcement, the mechanisms needed to secure connection on both ends must be in place. For instance, a remote or locally implemented form of authentication mechanism must be present, to be able to determine if the entity on the law enforcement side of the communication, is authorized to receive data output. Without authentication in place this program will not satisfy [CWE20a], by not keeping critical functionalities available to only those who can provide the proper credentials. For instance, keeping the system open to anyone would allow unauthorized use of collection of data, calling of actions and declaration of formats for use on data. Additionally, in order to implement an authentication mechanism, this system must also be capable of restricting authentication once a number of verification attempts have failed. In [CWE20b] the requirement is that once a number of attempts at authentication has been met with no success, the system should be able to detect and deter any further attempts for a amount of time. One proposal would be to implement a system in which three failed attempts will lock out that entity from authenticating for 30 minutes, while generating a ticket

about the failed attempts for logging. Furthermore, if these authentication credentials were to be linked to a user or email address, it would be appropriate to notify that individual about the incident. Authentication will be an important element that has to be considered in this project, because of the nature of the information contained in this formatter should not be available to outside observers. Another important factor that also should be discussed is on the distribution access and roles in this program.

7.0.8 Code security: Access control issues

In predisposing private assets without the use of proper authorization based on credentials, the confidentiality and integrity of the events and data from smart devices becomes compromised. Authorization is the process of checking if an entity on computer system is allowed to use predisposed assets available on that system. For instance, if there are actions available in the program that can only be used by personnel higher up in an organization, then there has to be a mechanism that prevents employees of lower rank from using these actions. Data access is another instance in which authorization must be enforced upon, given that some data might be classified as data that requires security clearance to view. Just as with the lacking implementation of authentication, this feature has also not been addressed in this program. The vulnerability definition in [CWE20d] infers that a system will be at risk if there is an absence of proper access control and authorization in a computer program or system. With no such mechanism in place, the resources on that system will become available for anyone who can connect to it. Moreover, an implementation of an authorization system should not violate [CWE20e], by improperly allow access to resources to those who have insufficient clearance. For the sake of this program, a possible solution would be to add an existing mechanism on to the existing system and declare that actions, formatting and handling of data must require proper clearance.

7.1 When one link fails

In relation to the holistic nature of security as hypothesized in the SBC model, when any of these requirements were to have failed at some point - i.e. its security is compromised - then the security of the other classes would have worse security as a result. This security measuring model in of itself suggests that security as a whole within an organization is only as good as the security of its components, whereas if one is weak, then the rest will be weak as well.

An example, if the integrity of the data that arrives is compromised because a cybercriminal is able to write erroneous data to the device, thereby effectively rendering it unreliable. Then the impact of the compromised integrity would affect other classes. For instance, erroneous data arriving on a formatting server could raise a false positive alarm and then the police would respond to an incident that did not take place at all. Alternatively, if the input were manipulated to make devices not broadcast an anomalous sensor reading, then the police would not be responding at all, leaving the perpetrator to do carry out their deeds.

For the people who are involved in an incident the improper response from police, due to a

misappropriated configuration of a sensor profile, could lead to victims being hurt. Alternatively, if the manipulation of events obscures the information that is being obtained by police, then the police could end up harming innocents as a result of misconstrued evidence which making law enforcement perceive someone to be dangerous and thus they could anticipate that they have to handle the situation using lethal force. Moreover, police's safety could be put at risk if misinformation were to be spread by a malfunctioning system.

For individuals, which to the police, are not considered Persons of Interest, should not have their device's data handled further than this program at all. All data that arrives on a computer that runs this program with the intent to filter out benign from anomalous should be discarded if the data is not deemed relevant. This implementation would be in place to protect the sovereign citizens right to privacy. If this system were to be circumvented, thus granting access to an adversary, then any promises to uphold the protection of a person's privacy rights cannot be guaranteed.

7.1.1 Summary

There are many considerations that one has to address when a system is to be introduced, which would disrupt the way an industry conducts their routines. Law enforcement and citizens must be secured with physical, legal and ethical needs in mind. If not, this system would cause more harm than good and therefore would not be adopted into a smart city.

Many of the problems in relation to legal and ethical standards could be infringed if the technical aspects of a application is not properly maintained. If a program is not properly configured and regularly updated to address security issues that emerges with time, there is no guarantee that compliance with laws and upholding ethical concerns to be achievable. Furthermore, the SBC assumes the threat to the security of one component, to a threat to the security of all components. If one link in the chain breaks, then the rest of the chain will lose its structure, and everything will fall apart.

8 Evaluation

8.1 Limitations

This project comprises of two methods of communication, the first one is Message Query Telemetry Transport, which is the one utilized by the TTN Api hook, while the protocol is based on websockets and is implemented through Golang's standard library. There are a multitude of different protocols that are used in IoT, but this evaluation will focus on a few implementations of formats and their preferential protocols. The reason why this evaluation focuses more on assessing its own performance against protocols, rather than assessing the formatting of these frameworks is that measuring data value to a community would be subjective. Assessing the value of data as evidence for police, in comparison to the value of medical data for physicians or financial data for a stockbroker, is that there are uncertainties as to how such a value could be measured. Values of data has that inherent nature of being a metric that has no definite and established definition of what could be considered valuable to a profession. In [ADJ14] it was discussed that an evaluation that bases itself on metrics that are inherently subjective, are metrics that are prone to be established on biased terms. In Law Enforcement, such a bias would exist in conjunction with how data is perceived to be to the individual investigator and field operator. For instance, while forensics have well-known sources of data that could be used as evidence, phone-call history, chat messages, video surveillance footage, server logs and so on. The fact still remains that other types of data from lesser known sources and that are infrequently sought to be of use, could be applicable in some instances, while in other cases it does not. Intrinsically the value of data cannot be measured due to its subjective nature, while intrinsically this subjective metric would make the results of such a metric in one domain different from another. The candidates that are chosen to be used in this evaluation are based on research conducted in the area of creating a formatting framework for IoT data for various purposes. The target of these proposed frameworks is their use of Application Layer IoT Protocols and these will be used to compare against the protocols used in this program. Since this program uses two different types of protocols, which are applied in two different ways, the testing done on these cannot be equivalent for all metrics measured. The measured metrics in this case are in part based on performance measuring of the program and the reliability of the protocol in terms of being able to handle large amount of traffic. Last discussed metric in this chapter is a few comments on the security of this program, with respect to this framework and other protocols.

8.2 Application layer IoT protocols

There are a multitude of different protocols that can be used in transferring information between endpoints. [BTS20] proposed a unified data framework for electronic health devices. Their proposed framework uses Contrained Application Protocol (CoAP) to tranfer data to and from the program. Another project that uses CoAP was proposed in [Chi16] and later developed further in [MBB18]. Their introduction to a formatting standard uses CoAP in conjunction with ThingSpeak API and Mobile Crowd Sensing. Other application layer protocols, such as Message Query Telemetry Transport (MQTT) is proposed in [Pra+18], when they were implementing an intelligence gathering framework for a smart city. When these frameworks are evaluated in this chapter, the main target for their evaluation will be on the protocols that they use. The projected results shown in this chapter based on protocol evaluation will therefore be assessed using metrics evaluated in research, with respect to these protocols. But in order to test the proposed framework against other protocols, a test scenario for this application must be created.

8.3 Testing framework

In order to show that this program is capable of handling a multitude of requests at once, a test scenario were created to demonstrate the overall power of this application. Since this program uses a combination of Websocket and MQTT, where both protocols are dedicated for retrieval of data from different sources, the testing has to be done separately on these two protocols.

Device	Type	Protocol	Data
Arduino MKR 1300	Hardware	MQTT	Humi/Temp (float64 x2)
Lightswitch	Software	Websocket	On-state (boolean)
Light bulb	Software	Websocket	RGB (int, int,int)
Code pad	Software	Websocket	key 1-4 (int x4)
Temp sense	Software	Websocket	Humid/Temp (float64 x2)

Table 2: Tested devices

The first test conducted on this program is a simulation of scenario where there a is high volume of activity at once. For this test, a function running a concurrent Websocket was implemented with the goal of observing what would happen if a large pool of devices were to communicate with a single host at once. Additionally, a second function were created that automatically generates a pool of semi-random generated devices. These devices can be assigned one of the following four test devices: Temperature/Humidity Sensor, Code-pad lock, LightSwitch and Light bulb. The first device is a temperature and humidity sensor that are configured to

broadcast random sensor readings to the program. The second device is a simulation of an output keypad entry that a user would hypothetically enter into a keypad to unlock a door. Third device is a light switch that can either have the value of on or off to indicate the state of the light switch. The last of these devices stores the color values of a colored led light bulb, thus simulating the same values broadcasted by a smart light bulb. What each of these devices are assigned is solely dependent on a random value that are assigned upon the initialization phase of each device. In total there will be approximately *2000 devices* created in this test scenario and for each of these devices a respective function will spawn a goroutine to instantiate a websocket connection to the main program's socket instance.

One important factor to take note of is that both the simulated client and server are run on the same host. In other words, these measurements do not take into account the extra time it would take for traffic to travel over a plethora of additional gateways. Moreover, due to the multitudinous amount of performance spent on creating new devices, randomly assigning new data to it and send it over to the server, there will be some additional overhead to the measuring of the program's performance. Another important thing to note about this test is that the overall number of concurrent connections are limited because of the latency stimulated by using a single host machine for the testing. Therefore, a 1 second sleep penalty is enforced on all running devices, right after the connection phase. This is deliberately done to reduce the chance of the server not properly setting up a connection in time before the device sends its data.

The second test cannot be done in the same manner as the first test that uses websockets. The MQTT protocol that is being used in this program is used in conjunction with a API that limits the connection bandwidth between a client/device and the cloud. For that reason, the total number of devices that can exist on a cloud application is limited to a 100 devices. However, as the number of devices increases, the latency of response from the server increases as well and therefore the reliability of issuing uplink messages from simulations becomes less feasible. For all intents and purposes, to test the performance of the MQTT is less conceivable so due to the limitations of the underlying service. In order to address the aspects of MQTT that cannot be measured directly in this program will be assessed using previous measurements conducted in related research.

8.3.1 Performance testing

In order to conduct a performance test on this program, the proposed testing program will count the overall time taken by the program to finish a single execution cycle. Since this program does not implement counting of all operations conducted across all functions in all files, the best option in this case is to observe the total execution time. As described above, the total requests made to the server from the client side is approximately 2000 devices, with some being expected to not send information, due to the limitations put on the current testing environment. In order to test this program properly, a single cycle (with 2000 devices) are repeated several times over on the same connection, to see if the duration skews towards lower or higher value in a different

cycle.

```
1        start := time.Now()
2        ...
3        currentTest.ServerTime = time.Since(start)
```

Listing 18: Time Difference calculation

The functions shown in listing 18 illustrates how the start and end times of the program are captured by the test cycle. These are placed in the program to set the time stamp before the networking sequences are called and one after the sequences are called.

Test	Bytes(Serv)	Bytes (Dev)
1	130986	2.1778662s
2	135021	2.1551524s
3	137697	2.1518888s
4	129978	2.1647948s
5	132836	2.1436297s
6	128906	2.1359591s
7	133325	2.6300981s
8	138069	2.1260972s
9	129365	2.5924328s
10	132910	2.5893287s

Table 3: Time tests results

As shown, in the table, the duration recordings on ten execution cycles have similar values to each other. There are a few cases where the execution of the program takes anywhere from 0.5 to 1 second more than the values here, and this could be attributed to the volatile nature of the go routines that are not entirely running in parallel. Moreover, the added penalty time imposed on all the 2000 devices created, the additional 1 second would also be accounted for here. It cannot be counted off for the time being, as the nature of this delay is to make sure that the program does not crash, due to a limitation set on the total amount of concurrent connections allowed in go networking. Measurements of time efficiency of protocols, such as

MQTT and CoAP is demonstrated in [KCK18]. In this paper, the researchers attempt to measure the time of sending and receiving packages with these two protocols, among others. The testing involved sending packets of differing lengths and quantities, ranging from 25, 100, 250, 500 and 1000 bytes in length and at quantities of 5, 10, 20 and 50 consecutive packets. Using the measurements on the packets that are of the similar size to the packets generated by this project, such that the compared results are more appropriate. The generated packets in the program has the respective byte sizes.

Data entry	Bytes(Data)
Code Pad	75
Light Bulb	86
Light Switch	53
Temp/Humid	91

Table 4: Default byte sizes of non-randomized data entries

The best option for a comparison would be to measure the packet timings for packets of 100 bytes in length. Could also be feasible to use the 25 bytes as well, but for simplicity's sake, the 100 bytes packets will be the chosen packets to measure. However, from the results shown in this article, the time difference between 100 to 250 bytes is not that significant, nor is the time threshold between 25 to 100 bytes that significant different. However, this is only the case for MQTT and CoAP. In [KCK18] the average time for these packets 0.24 seconds to 0.39 seconds for 100 byte MQTT packets at a quantity range from 5 packets to 50 packets. Meanwhile the time range for CoAP is between 0.23 seconds to 0.34. As shown there is a minor difference in speeds when juxtaposing these two protocols alongside each other. According to [PMY19] the CoAP protocol is based on User Datagram Protocol (UDP), which does not care for ordering of arriving data and puts less stress on reliability of the messages to arrive. Its lack of Acknowledgements (ACK) of transferred data makes data arrive faster on the opposing end-point. [1] Now, the MQTT protocol does require that acknowledgment messages to be fed back to be able to ensure that packets do arrive as intended and this is slightly taxing on the overall efficiency with respect to duration. Moreover, these authors also demonstrated that regular HTTP protocol performs closely to MQTT on small packets, mainly due to the fact that MQTT does not employ additional threading and fault handling features. When measuring these protocols for performance times when sending a 100 byte packet 5, 10, 20 and 50 times over.

[1] Granted that the packet reaches its destination

Protocol	5	10	20	50
MQTT[KCK18]	0.23	0.26	0.27	0.39
CoAP[KCK18]	0.23s	0.25s	0.25s	0.35s
HTTP[KCK18]	0.2s	0.3s	0.41s	0.79s
UIOT-FMT[Sel20]	0.0011s	0.0022s	0.0034s	0.0058s

Table 5: Protocol time comparison

The current table shows the results shown for the research results discovered in [KCK18]. The results show that the time spent sending and receiving packets takes much longer to do for all of the other packets contrary to resulting average time gained in this thesis. The way average was calculated in this demonstration was done by summing the 10 results together and then divide by the number of cycles. Furthermore, the value of 1 were subtracted from the average to adjust for the 1 second penalty that was implemented to prevent goroutines from racing to finish too fast. Much of the reason why this framework works more efficient is attributed to two plausible causes: the first reason for this is that the testing does not implement any form of assurance that information is sent, thus allowing for dropping packets. The other attribution to why these performance results comes out with such a low number could be that this program actively perpetuates the use of concurrency for all methods to reduce overhead on the main thread. While it is clear that the use of goroutines pose some advantage in the area of efficiency and lower latency, the question would then be if this is a safe way of transporting and handling data on a day-to-day operation. It might appear to be more efficient in some aspects, but this program does not take into account the packet re-transmission times, which is prevented by only allowing one transmission attempt. The next metric that should be addressed in this research is on the consequences of doing packet handling in such a manner.

8.3.2 Reliability testing

As discussed previously in chapter six, with lacking sources of information about device fingerprints present for the program to use, there is little information that would enable this project to employ more automation. Therefore, this program cannot measure for this What the program can measure, however, is the extent to which it is capable of maintaining information that is fed through it. This test will therefore be conducted on the overall capability of the program to receive packets from the clients and how much of the information will not make it to the client. The first limitation imposed upon the program is a policy to which the packet is not re-submitted to the server if it failed to send. If the packet for any reason fails to send, the amount it presumably had sent is logged on the client side and stored. On the server side, all packets that has been received by the server is stored in a buffer with the number of bytes received.

After each of the components in the program has terminated their cycles, the total number

of bytes logged on each side is calculated to find the percentage of loss.

$$Packetloss(p) = (\frac{Device(P) - Server(p)}{Device(p)}) \times 100$$

(8.1)

This formula will measure the percentage of loss in bytes occurring in the program. The formula takes the fraction of the counted bytes on the server and the counter bytes from the devices. The way that these bytes are counted is done through the *net.write* and the *net.read* command, which both have a *bytes processed* return values. When network connections send and receives these bytes, the functions that handles these operations would confirm the amount of data sent and this is how the exact amount of data was found.

Test	Bytes(Serv)	Bytes (Dev)	Loss	Percent
1	137496	139647	2151	1.54031%
2	139517	140353	836	0.59564%
3	136745	137905	1160	0.84115%
4	130142	132780	2638	1.98674%
5	133362	136300	2938	2.15553%
6	133127	133519	467	0.34976%
7	127592	129836	2319	1.78609%
8	131885	134884	2999	2.22339%
9	133955	135656	1701	1.25390%
10	130923	135204	4333	3.20478%

Table 6: Packet loss results

Listing 6 shows the resulting packet loss experienced over ten consecutive executions of the program. The number of devices is set to the default test value, with approximately 2000 devices broadcasting data to the server. No confirmation of delivery is supplied to the devices, nor is there a mechanism to enable a device to resend the data if a failure is detected. The only check conducted is the sending mechanism on the device side of the communication, which requires that the dispatch function supplies the amount of packets that were sent. The loss rate is calculated using the above-mentioned formula. It is shown that the current loss rate varies between the 10 cycles. In some cycles, the overall loss rate is below 1%, while in other cases

the loss rate of the program can reach as high as 3.2%. This rate of packet loss in a testing that simulates a traffic burst over a short amount of time is deemed reasonable. In [Bor+] it was concluded that if a system manages to maintain loss rates at a level of 3.6% or lower, then that system will be able to serve its purpose with minimal impact to its Quality of Service (QoS). The loss rate seen in this scenario could be attributed to the fact that the amount of traffic is large and because of the lack of re-transmission mechanisms, which causes the jump in loss rate.

	MQTT[Shi+13]	**CoAP[SK19]**	**HTTP[Bor+]**	**UIOT**	**UIOT^2**
Loss(%)	0.20%	7.24%	3.6%	2.18%	0.042%

Table 7: Packet loss result

According to [SK19] protocols such as CoAP that are based on UDP have a tendency to lose more packets when the amount of ongoing traffic is too high. They demonstrated this in a similar test scenario to the one demonstrated in this paper. They showed that the loss rate on different topologies that use CoAP will experience high rates of loss when the volume of traffic increases. This can further be seen in [AB17] where Packet Loss Rate (PLR) is directly affected by the amount of traffic on the destination. Their For the MQTT protocol, the same correlation can be drawn between reliability of packets and their timeliness. [Shi+13] tested different 4-way handshake methods in MQTT to see where the reliability of information will change when activity on a connection increase. They showed, that when enforcing a regular 4-way handshake, the overall packet loss decreased slower than what it would when less strict handshakes were used. The percentage on a regular 4-way handshake on a wired connection to be around 0.20% when the load is around 16000 bytes. When calculating the average packet loss rate of the program, using the same method that generated the cycle values in table 6 amounted to a value of 2.18%. However, with the given discovery that large traffic generates lower successful transmission, the question would be what would happen if the delay penalty were to be randomized, instead of being a constant.

```
1 time.Sleep(time.Millisecond * time.Duration(Utils.RandInt(3000,
2     10000)))
3     //Sets a random time penalty on each device that are
4     //running in a goroutine
```

Listing 19: Random Time penalty

In listing 19 it is proposed to modify the current testing framework, by changing the persistent

penalty value that is being used on goroutines, to a value that is non-constant. The idea here is that when a random time delay is being introduced the distribution of which these routines are to proceed is more scattered. This would hypothetically in effect make the server experience less simultaneous traffic and become more capable of serving the devices as a whole. By introducing a randomized additional time penalty to each and every device the overall loss of packets is reduced and resulting in an average packet loss rate of *0.042%*. Moreover, the overall number of devices could be increased to 5000 devices with this random delay. While 10,000 devices were also feasible, but with some cycles crashing on occasion. The changing of this value have shown that while the reliability of the program can be improved, but the overall speed is affected as a result.

8.4 Summary

This chapter showed that it is possible to simulate a large number of devices for use in format analysis and data extraction on this framework. The initial testing was somehow unbalanced, thus giving more priority on the efficiency of performance. This could be seen in the way that the delay was set to a constant, thus only alleviating the problems of crashes and lost packets. However, when the random delay was introduced the amount of faulty devices were significantly reduced, to the point where the overall reliability improved. This would of course be at the cost of performance in the program. In order for this program to flourish in terms of the metrics measured in the evaluation, there has to be conducted further testing. These tests must be done with the intention of adjusting the balance of reliability versus performance, such that one metric does not affect the quality of the other.

9 discussion

The law enforcement personnel and forensics are not optimal, with several issues that they face on their job. What is apparently more present in the field operator's agenda is the problem with safety hazards relating to their occupation's tasks, such as the problems with how many officers experiences physical harm to their bodies as a result of the lack of sufficient information to be able to make a sound decision. On the other hand, when it comes to the mental hazards in this profession as a whole, there are prominent concerns to be had with respect to its presence in both the field operators and the forensic team's workday. However, what sets these two categories apart when it comes to experiencing the degradation of mental health is attributed to the source of the trauma. With Field operators, the root cause of their stress is experienced through encountering difficult and endangering situations in the field, whereas forensics suffers from the problems of encountering unsettling material. Law enforcement's main focus would be to obtain the knowledge necessary to reduce the risk of being caught in a difficult situation, while increasing the likelihood of being able to resolve it, with as little collateral damage inflicted as possible. Whether or not this can be achievable through the use of IoT technology is a question that awaits an answer. What limits one from answering this question is the limitations of access to this evidence. For instance, with more access to a platform that harnesses organic and real data streams from IoT devices, the likelihood of grasping the value of IoT sensors for use in crime fighting will become more conceivable. This is of course granted that the data obtained can be used for creating scenarios that would simulate actual indicators of a crime, because without these indicators being established, there is no clear definition of what data values would solicit a system to raise an alarm or when it should not raise any alarms. In forensics, however, the main problem would be on how the data were to be obtained once a crime has been detected. Because of the limitations set on acquisition of data from, device level, the intermittent layer and the cloud layer, the forensic investigators would have to subpoena service providers to access customer data on the cloud or invent new methods of obtaining evidence from devices on the intermittent and device layer. When it comes to the latter, the problem could be further exacerbated by the sheer fact that IoT encompass a plethora of different devices.

One of the issues with conducting any form unification on a populous of any kind, is that when the diversity of said populous increases, the additional required steps needed to take into account the majority would increase as well. Chapter 5 showed that there exists a large array of devices that can cater to a whole host of various aspects in Smart City. For instance, creating a format for all devices encompassing the city as a whole is not feasible with the current landscape of technological diversity and a saturated market. One proposed method would be to

manually create format recipes based on the devices that are encountered. Chapter six covers the underlying problems with the reliability of data in terms of providing accurate localization and identification of entities in a crime, but it also discusses the problems with how devices are categorized. Mainly to be able to create a library of format to automatically assign to a device, there must be some form of information supplied by the device, in order for this to be achievable. Unfortunately, the discussion as to what this type of information is and if it is a consistent identity found on every device, is a problem that is yet to be solved. In order for the program to be able to adapt to a diverse IoT landscape the problem is that there must exist a defined ruleset which are used to conform the data when it is not of the correct type. Chapter six also proposes that automatic type checking is enforced on the data when it has arrived, such that no misappropriated input is being processed and dispatched. Whether or not this is an appropriate is dependent on the person who are using it, as their intentions could require that data is outputted in a particular manner. For instance, a data scientist would require that data output is done in a particular integer/float type for use in machine learning. While on the other hand, some intentions for use of this program might not require any operations to be done to the data at all. Another topic equally as important is the ability to handle a vast amount of data at once.

Scaling the application to be able to handle as much information as possible is very important for this project, given that the area of application will contain a large number of devices that needs to be processed. In the evaluation it was identified that the problem with the proposed Websocket server is attributed to an unbalance between the performance and accuracy. When the priority is set to favor performance, by making the program execute as fast as it can without interruptions, the reliability of that program would be severely reduced. An increase in reliability was achieved, but at the cost of performance, as the methods used to decrease lost packets involved setting a higher threshold on the total time that the devices had to wait. However, with the introduction of added delays at a random distribution, the overall load on the server was reduced. With a reduced load, the ability to scale up the number of devices was possible, which increased the number of total devices by a factor of five. An alternative proposal to handle the reliability problem would be to implement an acknowledgement of received package. For instance, instead of assigning random time delays between 3-10 seconds, the overall wait threshold could be reduced if the system has a way to verify transactions and re-transmit lost packets. However, this would only reduce some constraints on the performance, as the requirement fro re-transmission would require one or more additional dispatch attempt. With a proposed application that creates a format for use in a society in need of improved crime fighting, there are inevitable impacts that must be discussed.

Making a concept that allows and perpetuates the idea of establishing a mechanism that can provide insights of a person's daily life is considered equivalent to surveillance. While this system could be considered to be similar to a surveillance system, the underlying idea is that there should be no human controllers in place to observe the data that is being processed. The

controller of which this program was developed to be a part of is intended to be autonomous and requires no human interaction. The only intended insight that human should have is when there is a clear indication of a crime taking place. But just as how the data limitations are caused by limitations on its access and fingerprinting capabilities, the problem with this autonomous controller is that there is no established recipe for how it will observe data and derive meaningful information from it. What can be a value that would distinguish a normal room humidity from one that could indicate a house fire? Additionally, what more information is required to further prove that an incident is in fact legitimate? An autonomous controller must be able to determine the nature of all the data that is processes, such that it can carve out the anomalies from the indicators that it discovers. But this will not be a complete solution to prevent others from gaining insight into the data. If the framework is not properly secured against attacks, the outcome may lead to scenarios where attackers might obtain access to the sensor data. A compromised security would mean that someone would gain control over its resources, and if this were to be the case; the problem would be whether or not this system still will not have to be considered a surveillance system. An alternative definition of this system, when there are security concerns, such as lacking authentication and weak access control, could be a computer system with the plausible potential of becoming a surveillance system. While it no longer is the operators who observes the data, the inclusion of outside forces would still make it a surveillance system, even if it was not intended to be such a system.

10 conclusion

This project introduced a precursor to a system that sets out to become an IDS system for a smart city, where the its goal is not to detect compromise to a computer system, but rather the compromise to its inhabitants. In developing such a precursor program, it was shown that it is feasible to achieve data fusion with the help of functions that ensures that the data is formatted to an appropriate output. However, problems pertaining to indexing new devices and fingerprinting to attribute existing formats is not yet feasible. The outcome of this system is proposed to be an aid used by law enforcement to reduce the chances of harm, while increasing the probability of crimes being prevented. However, it is only theoretical that law enforcement can benefit from this format and more research must be conducted into establishing an IDS mechanism, train this mechanism with the appropriate rules to teach it to detect crimes and then test this on the data.

11 Future research

11.1 Ethical review of privacy facilitation in automated surveillance

As described in [YSK20], the proposed detection system draws some parallels with the fictional Artificial Intelligence surveillance system, from the TV-show Person of Interest. Both systems are based on an automated program, to which only the anomalous incidents get reported to the operators, leaving all other benign information a secret. In this research question, the student should discuss the justification for deploying an intrusive automated surveillance application in a smart city and why it would be less damaging to one's rights to privacy.

11.2 Application security

By collecting, analyzing and aggregating data from a multitude of devices, it is obviously a security concern that malformed data can cause issues with the program in such a way that it misbehaves or reveals information to the public. As a proposition for a research activity to address such this problem, a research project that is based on attempting to fuzz the program with the goal of circumventing it should be conducted to discover possible weaknesses.

11.3 Format profiling and standardization library

To figure out a way to procure and maintain a library of data format profiles that could be used by the program as a means of detecting and recognizing what it should do when encountering a specific device. The idea behind this research activity is to stimulate a debate on the feasibility of establishing a library of proprietary data formats, that includes the common serializations, such as XML, JSON and raw binary streams. In such a manner that the incident detection system can automatically parse the data, based on what profile it selects.

11.4 Profiling Indicators of abnormal events

When developing a detection system, it is also important to establish a baseline as to what is considered relevant for raising an alarm. Given that this project is to establish an automated system to detect anomalous incidents that are detected through smart devices, it is also important to define what is considered as anomalous. In undertaking this research problem, the researcher has the opportunity to discuss whether the next iteration of this detection system is to operate as a signature based or an anomaly based detection system. For instance, should the system be operating with a baseline for accepted behavior or should is index the types of inappropriate behavior?

11.5 Integration with Machine Learning

In order to integrate IoT data with Machine Learning an output must be configured, such that the data can be offloaded onto a system that learns or classifies entities. Such as using Convolutional Neural Networks (CCN), the data output must be configured to output the data as an output that conforms to the input expected by a system that is running a CNN.

Appendices

A Installing the program

A.A synopsis

To run this program there is a few steps required to setup the environment. This program is primarily developed in Go and therefore the Go programming language is required to be installed, in order to run the source files.

Go projects have some benefits in terms of easy installation of internal dependencies. If for instance a go project uses a lot of standard or user created libraries - referred to as packages - the installation of these will not be required.

The following steps in this setup tutorial assumes that the user has access to a reasonable modern computer, that is running on a 64-bit architecture with Ubuntu installed.

The machine used to recreate these steps had the following hardware:

1. Intel Core i7-6820HQ 2.7GHz
2. 24gb physical memory
3. 64-bit arch
4. Windows 10 Professional (Ubuntu running on Docker)

A.B Requirements

A.B.1 Installing wget, unzip and git

In Ubuntu, the first step to start installing the pre-requisites is to do an update of the package manager:

```
root@hostname#  apt-get update && upgrade
```

To install *git* on Ubuntu type:

```
root@hostname#  apt-get install git
```

and for *wget* the same command can be used as follows,

```
root@hostname#  apt-get install wget
```

Lastly, we need a archive manager that handles zip archives. **Use the apt package manager to download unzip.**

```
root@hostname#  apt-get install unzip
```

A.B.2 Installing Go

Now that wget and git is installed on the computer, the next step in the installation process is to install Golang on the computer. Use the wget command to download the source tarball for Go.

```
root@hostname#  wget \\
    https://dl.google.com/go/go1.14.3.linux-amd64.tar.gz
```

The link listed, points to the GNU/Linux source archive that is hosted on Go's website. Next the tarball has to be unpacked and placed into a folder that is recognized by the systems local environment. The best location to unpack the library is in the */usr/local* folder.

```
root@hostname#  tar -xvf go1.14.3.linux-amd64.tar.gz
root@hostname#  sudo mv go /usr/local
```

Once the main go files are placed in the appropriate folder, the next step is to set the system variables. A sytem variable is a reference that the bash shell uses to keep track of session variables and executable commands. In order to use Go from the command line, the appropriate environment variable must be set. The first variable that must be set is *GOROOT* which is the main folder of the go source code. If the archive was unpacked to */usr/local*, the path to assign to *GOROOT* would be:

```
root@hostname# export GOROOT=/usr/local/go
```

In order for Go to store local and imported projects and compile binaries, it requires that a folder is created to store all of these files. The environment variable *GOPATH* will point to the folder that Go will use for the compilation of programs and the importing dependencies from Github. This tutorial will create a folder called go in the home folder and set this to be the *GOPATH*.

```
root@hostname#  mkdir -p /home/username/go
root@hostname#  export GOPATH=\$HOME/go
```

The last step to install go is to assign the two newly created varibles into the local *PATH* variable. The important thing is to add the bin subdirectories of each respective variables, as these will allow for the execution of Go binaries and binaries that you will inevitably compile in go.

```
root@hostname#  export PATH=\$GOPATH/bin:\$GOROOT/bin:\$PATH
```

From here go should be available from the bash command. It can be tested by entering the following command into the command line:

```
root@hostname#  go help

"Go is a tool for managing Go source code.

Usage:

        go <command> [arguments]"

...
```

If the command returned the text beneath, then the Go library has been sucessfully installed on the system (So far, so good!). The next step is to download the project files. Before acquiring the files, the normal way of acheiving this would be to use the *go get -u* command as shown below:

```
root@hostname#  go get -u   \
    github.com/matsse/SmartPolice-Interface
```

Unfortunately, for the time being this repository will remain private. The alternative method to obtaining the source files will have to be done using Dropbox. First create the following folder in *GOPATH*.

```
root@hostname#  mkdir -p \
    GOPATH/src/github/matsse/SmartPolice-Interface
```

The reason why this directory was created is because this is where go would have stored the repository if it were to be downloaded with *go get*. **Next we change into the directory that we just created.**

```
root@hostname#  cd
    GOPATH/src/github/matsse/SmartPolice-Interface
```

In this directory, we have to download the necessary source files. Using wget and the *output (-O)* parameter and the *–max-redirect=20* parameter we can download all files used in the repository.

```
root@hostname# wget --max-redirect=20 -O Out.zip
"https://www.dropbox.com/sh/wuihdgni4m
zufxd/AABcyZebhdHqe8VpYZXGrP8La?dl=1"
```

Once *wget* has finished downloading the folders, the next step would be to unzip the archive into the same folder where it was downloaded. We can use unzip to unpack the archive.

```
root@hostname# unzip Out.zip
```

If everything went well, the current directory listing should show the following input:

```
root@648f9ad7bfe5:~/go/src/github.com/matsse# ls
 Core  DataOutput  Formats  Protocols  README.md    Tests
 go.sum   run.cmd Data  Devices  Out.zip  Providers  Screenshots
 go.mod  main.go
```

B Main program

B.A Overview

After we have followed the guide in appendix A in this thesis, we would have all of the necessary files required to run the program. This program is entirely written in the Go programming language and comes equipped with both self authored code, as well as user created and standard libraries.

B.B Core Components

This program comprises of three primary components to which the program relies on during runtime. The core components of this program involve the formatting of data, management of devices and handling of formatting.

B.B.1 Formatting

Formatting of the files requires some files to initiate the formatting upon start of the program. Furthermore, the program must allow the user to define their own formats.

Name	Type	Status	Desc.
Data/Formats	Directory	Use	Storage space for formats
Core/format.go	File	Use	Loads and Creates format files
Core/Actions.go	File	Deprecated	Initially used to read actions
Core/Actions/	Directory	Use	Stores action files and parsers

Table 8: Components used to format data

In table 8 the entities used for managing formatting is quite a few. The establishment of new formats are done in ***Core/format.go*** where the user can create new formats and store it in ***Data/Formats***. Upon the start of the program the ***main.go*** file will connect these files together and the user must specify the format command to access the format creation prompt. Originally, when formats were read by the software, they would have to go through the ***Core/Actions.go*** file, because this was the one who checked for *action* and *chain* keywords. However, this file has been now been replaced with other files. But ***Core/Actions.go*** it is still used for demonstration purposes, because it requires no argument specification. This file is therefore useful for demonstrating action calls, where the only argument used is the data entry itself.

B.B.2 Format handling

Name	Type	Status	Desc.
Core/Utils/functionSelector.go	File	Use	Selects the right action
Core/Utils/typedVariables.go	File	Use	Parses arg. data type
Core/Actions/FunctionAssembly.go	File	Use	Error checking and exec

Table 9: Components for handling

The handling components are the parts of this program that makes sure that the data is in order. ***Core/Utils/functionSelector.go*** is the program that selects the correct function for the action specification. This file is what is replacing ***Core/Actions.go***, but for the time being, this file is only used for actions that takes two or more arguments. Type checking is performed in ***Core/Utils/typedVariables.go***, where formatted types are checked when loaded into the program. Since an interface will automatically assume numbers to be be *float64* and arrays to be *[]interface*, this function will validate the specified type in the format against the actual type in the data input. ***Core/Actions/FunctionAssembly.go*** handles the type and verification functions of action functions that takes arguments. Type checking of each argument will be done in this file, as well as the ensuing execution of the action.

Name	Type	Status	Desc.
Core/Actions/math.go	File	Use	Math functions
Core/Actions/Cryptography.go	File	Use	Crypto functions
Core/Actions/FileIO.go	File	Use	File output functions
Core/Actions/Converter.go	File	Use	Data type converter

Table 10: Action files

The action functions are found in the files listed in table 10. The purpose for these actions are to transform data if there is need to transform it. For instance, ***Core/Actions/math.go*** would contain Arithmetic functions, such that a user can adjust data when needed. Although the test concept only has a few functions. ***Core/Actions/Cryptography.go*** was intended to be used for demonstration purposes, to show that it is possible to perform encrypton and decryption on data. An example of this is shown in the next appendix. The converting functionalities of this program can be seen in ***Core/Actions/Converter.go*** and writing to files is handled by ***Core/Actions/FileIO.go***. All of these functions are intended to demonstrate what the program could potentially do, but it does not represent a final product in any way, shape or form. Other functionalities, such as concurrent web socket communication with foreign computer

programs, serialization for machine learning and so on, requires additional functions ensure that the program runs smoothly.

B.B.3 Device functions

Name	Type	Status	Desc.
Providers/TTN.go	File	used	TTN cloud API
Providers/ThingSpeak.go	File	Deprecated	TS cloud API
Core/device.go	File	use	Device handler
Data/Devices	Directory	use	Device storage

Table 11: Data files

The file ***Providers/TTN.go*** and ***Providers/ThingSpeak.go*** are used to connect to cloud services. At this time, only TTN is used for the project, as ThingSpeak is considered an unreliable source of data. ***Device.go*** is intended to be allow for the creation of new device, load devices on startup and properly manage device information. All files generated by ***Devices.go*** is stored in the directory at ***Data/Devices***.

B.C Running the main program

To run the main program from the source directory that was created in Appendix A, the following command can be used:

```
root@hostname#  go run main.go
```

Once executed a prompt screen is presented to the user. The options presented will navigate the user to other functions in the program. Options that are available is shown in the following welcome screen:

```
    device)    Enters the device management interface
    format)    Enters the format management interface
    listen)    Listen for the data that is gathered
    options)   Adjust settings
    exit)      Exit the program
```

In the following sub sections, some of the basic operations will be demonstrated.

B.C.1 Create a new device

To create a new device the user can enter the command ***device*** and press return. The following screen will appear.

```
Please select one of the options:
    new)       Create a new device
    edit)      Edits an existing device
    delete)    Deletes existing device
    dat)       Dump raw data from devices
    list)      List all existing devices
    exit)      Go back to previous prompt
```

In this screen, the user have to enter ***new*** and press return. A prompt will ask for more information about the device. The first prompt will ask for the global name of the new device.

```
Device#   new
What is the device name?
```

This first name is used by the program to reference to that device. Next the prompt will ask for a the type of API to use.

```
What type is this device?

        TTN)    TTN LoRA device
        TS)     ThingSpeak.com
```

The current api that is available is the Things Network, so the only option here is to enter ***TTN*** into the prompt. Initially, this prompt will redirect the user to either of the cloud API files, where the user will be prompted for specific information.

```
What is the TTN application ID (appID)?
```

Here the user will have to specify the correct name of the application, to which the device is connected to. For instance, in the TTN console if the application name is ***huelight***, the user would have to enter this name.

```
1 What is the TTN access key?
```

Access keys are important for the program to properly communicate with the application and the API. This key can be obtained from TTN's console dashboard. An example of a key could be: ***ttn-account-v2.nIxlEQvAfmYB5W7OFFtj_mghjh8HnLcX7ebbKV5QgCs***

```
1 What is the device name?
```

The prompt will ask for a name again. This is not the same name as the one used in the application. This name corresponds to the name of the device set in the TTN console and used by the API to find the device.

```
1 What is the description of this device?
```

THis field is optional, it is used for instances where the user wishes to add some arbitrary information about the device that they are adding.

```
1 Please enter the Application APPEui in space separted hex values
```

An APPEui is an 8 byte array of hex values that the server requires to identify the device and application. To enter this value, the user must specify all 8 bytes in hex notation and separate them with spaces. An example of an APPEui could be: ***0x70 0xB3 0xD5 0x7E 0xD0 0x02 0xD9 0xFA***

```
1 Please enter the Device DEVEui in space separted hex values
```

Just as in the previous step, the user must supply the prompt with an 8 by space separated value. When all of these steps are done, the program will automatically save a file to the ***Data/Devices*** directory. From this point, whenever the application is re-executed, the JSON file with the device information will reload the device into memory.

B.C.2 Create new format

To create a format using this program, the main screen has an option called ***format***, which can be called by the user to open up the format creator screen.

```
1  What will the name of your format be?
2  NAME#
```

The first step is to give the format a name. and it is up to the user to decide what the name of the format should be. However, the use of special characters and spaces are prohibited.

```
1  Target#    key:temperature
```

After the name has be entered, the user will be presented with a screen that will inform about how to enter the correct information. The feneral overview of the screen is that the user can either enter a key or a path value. In the listing above, the key value is entered. It is important that these values are entedred with the prefix ***key:*** and ***path:***, before entering the target value.

```
1  Target#    path:root.values.temperature
```

If a path value is needed instead of a key, then the appropriate method will be to separate each keyword in the path with a period (.).

```
1  Target#    type:int
```

The next step in the format creation process is the type definition. The idea of this is to define a value that corresponds to the type of the data field. The screen that handles the definition of types provides a list of the accepted types for this program. Currently, however, the main program has focused on only a few of those that were listed.

```
1  Actions are functions that transforms data and manipulates it
2   to the users liking.
3  Do you wish to add actions to this format (yes/no)?
```

This page asks the user if they wish to create an action to add to the program.

```
action:function1(sint#12,sint#14)
```

To specify an action the keyword ***action:*** must be specified. It is important that the user specifies the correct name of the action and the arguments with their type. For the arguments, the user first has to specify the s-type, followed by a number sign (#). An s-type is an alternative keyword for type, where the intention is that argument types have their own names for their respective data types. For instance, the type definition ***int*** becomes ***sint***, ***string*** is ***ssint***, ***float64*** becomes ***sfloat64*** and so on. Other important type names that are included here are listed in the table below.

```
chain:convert2Int(self#).function1(sint#12,sint#14)
```

To specify chains the same steps taken to declare actions is required, but for each additional action added to the chain, a period is needed to separate the actions apart. Once this is done, the format screen will output the resulting format and its actions.

Name	type	s-type	Exsample
Reference	N/A	ref	ref#name
Self ref	N/A	self	self#name
Integer	int	sint	sint#13
Float32	float32	sfloat32	sfloat32#12.1
Float64	float64	sfloat64	sfloat64#12.1
String	string	sstring	sstring#hello world

Table 12: S-type values

C Running test files

This program has alot of modules added onto it over time and by that there was a need to create debugging tests to execute along the way to see how the program works with additional features. The following appendix will cover over the execution and the different test files that were used during the project creation. There are several files created during the development of the prototype. These files covers a specific aspect of the program with respect to the format's features and design. Thse files themselves does not provide much information, other than being used as testing to see if the program can be executed with the code that is in the file.

The abuse test file ***Abuse_test.go*** was intended to see what would happen if a file that contained a malformed action call would do to the system. To run this file the following command can be used:

```
go test Abuse_test.go
```

Testing of actions and chains can be found in the file ***Actions_test.go***, where the intent is to see if the program can recognize actions and chains from data input.

```
go test Actions_test.go
```

In the evaluation chapter, the file ***Benchmark_test.go*** was the test file that were utilized to randomly generate device data and send it to a local server. The command to run this test file requires two additional arguments to be entered. First the argument ***-count=1*** is needed to prevent the test from caching after termination, while the keyword argument must be replaced with either ***Time*** or ***Loss*** for the program to calculate the loss and time duration of the program.

```
go test Benchmark_test.go -count=1 ARGUMENT
```

The file ***Format_test.go*** is used to check for ***key/path*** values for the format and the type of the target.

```
go test Format_test.go
```

Testing the connection with the TTN cloud server can be done in ***TTN_test.go***.

```
1    go test TTN_test.go
```

Bibliography

[U.S. Const.] Constitution of the United States. 1787.

[US 68] U.S. Congress. Wiretap Act. 1968.

[Kow94] Stewart Kowalski. "It Insecurity: A Multi-disciplinary Inquiry". Doctoral Thesis. Royal Institute of Technology and Stockholm University, 1994, pp. 183–199.

[EC00] Anke Ehlers and David M. Clark. "A cognitive model of posttraumatic stress disorder". In: Behaviour Research and Therapy 38.4 (Apr. 2000), pp. 319–345. ISSN: 00057967. DOI: 10.1016/S0005-7967(99)00123-0. URL: https://linkinghub.elsevier.com/retrieve/pii/S0005796799001230.

[US 01] U.S. Congress. UNITING AND STRENGTHENING AMERICA BY PROVIDING APPROPRIATE TOOLS Washington D.C, 2001.

[VK04] Vijay Vaishnavi and B Kuechler. "Design Science Research in Information Systems". In: Association for Information Systems (Jan. 2004).

[Cro05] Terry Crowdy. The enemy within : a history of espionage. New York, NY, USA: MW Books Ltd, 2005. ISBN: 9781841769332.

[Y05] Shafranovich Y. Common Format and MIME Type for Comma-Separated Values. 2005. URL: https://www.ietf.org/rfc/rfc4180.txt%7B%5C#%7Dpage-1.

[Pef+07] Ken Peffers et al. "A Design Science Research Methodology for Information Systems Research". In: Journal of Management Information Systems 24.3 (Dec. 2007), pp. 45–77. ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. URL: https://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302.

[Gar10] Simson L. Garfinkel. "Digital forensics research: The next 10 years". In: Digital Investigation 7 (Aug. 2010), S64–S73. ISSN: 17422876. DOI: 10.1016/j.diin.2010.05.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S1742287610000368.

[Mul11] Sameena Mulla. "Facing Victims: Forensics, Visual Technologies, and Sexual Assault Examination". In: Medical Anthropology 30.3 (May 2011), pp. 271–294. ISSN: 0145-9740. DOI: 10.1080/01459740.2011.561820. URL: http://www.tandfonline.com/doi/abs/10.1080/01459740.2011.561820.

[Tel11] Phillip Tellis. Keeping Web Users Safe By Sanitizing Input Data. 2011. URL: https://www.smashingmagazine.com/2011/01/keeping-web-users-safe-by-sanitizing-input-data/.

[Shi+13] Shinho Lee et al. "Correlation analysis of MQTT loss and delay according to QoS level". In: The International Conference on Information Networking 2013 (ICOIN). IEEE, Jan. 2013, pp. 714–717. ISBN: 978-1-4673-5742-5. DOI: 10.1109/ICOIN.2013.6496715. URL: http://ieeexplore.ieee.org/document/6496715/.

[ADJ14] Alejandrina Aranda, Oscar Dieste, and Natalia Juristo. "Evidence of the presence of bias in subjective metrics". In: Proceedings of the 18th International Conference on Evaluation an New York, New York, USA: ACM Press, 2014, pp. 1–4. ISBN: 9781450324762. DOI: 10.1145/2601248.2601291. URL: http://dl.acm.org/citation.cfm?doid=2601248.2601291.

[Cen+14] Angelo Cenedese et al. "Padova Smart City: An urban Internet of Things experimentation". In: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia N IEEE, June 2014, pp. 1–6. ISBN: 978-1-4799-4786-7. DOI: 10.1109/WoWMoM.2014.6918931. URL: http://ieeexplore.ieee.org/document/6918931/.

[CIA14] CIA. "EzCheese v6.3 User's Guide". Langley, VA, 2014. URL: https://wikileaks.org/vault7/document/EzCheese-v6%7B%5C_%7D3-User%7B%5C_%7DGuide%7B%5C_%7DRev%7B%5C_%7D%7B%5C_%7D2014-01-07/EzCheese-v6%7B%5C_%7D3-User%7B%5C_%7DGuide%7B%5C_%7DRev%7B%5C_%7D%7B%5C_%7D2014-01-07.pdf.

[Cra+14] Sarah W. Craun et al. "A Longitudinal Examination of Secondary Traumatic Stress among Law Enforcement". In: Victims & Offenders 9.3 (July 2014), pp. 299–316. ISSN: 1556-4886. DOI: 10.1080/15564886.2013.848828. URL: http://www.tandfonline.com/doi/abs/10.1080/15564886.2013.848828.

[Wei+14] Yanhao Wei et al. "Credit Scoring with Social Network Data". In: SSRN Electronic Journal (2014). ISSN: 1556-5068. DOI: 10.2139/ssrn.2475265. URL: http://www.ssrn.com/abstract=2475265.

[Gau+15] Aditya Gaur et al. "Smart City Architecture and its Applications Based on IoT". In: Procedia Computer Science 52 (2015), pp. 1089–1094. ISSN: 18770509. DOI: 10.1016/j.procs.2015.05.122. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050915009229.

[Wan+15] Meisong Wang et al. "City Data Fusion: Sensor Data Fusion in the Internet of Things". In: (June 2015). arXiv: 1506.09118. URL: http://arxiv.org/abs/1506.09118.

[Bea16] Tod Beardsley. R7-2016-07: Multiple Vulnerabilities in Animas OneTouch Ping Insulin Pump. 2016. URL: https://blog.rapid7.com/2016/10/04/r7-2016-07-multiple-vulnerabilities-in-animas-onetouch-ping-insulin-pump/.

[Chi16] Alain Di Chiappari. "A Collaborative Mobile Crowdsensing system for Smart Cities". Masters thesis. UNIVERSITA DI ' BOLOGNA, 2016.

[CIA16] CIA. "Brutal Kangaroo Program Drifting Deadline v1.2 User Guide". Langley, VA, 2016. URL: https://wikileaks.org/vault7/document/Brutal%7B%5C_%7DKangaroo-DriftingDeadline-V1%7B%5C_%7D2-User%7B%5C_%7DGuide/Brutal%7B%5C_%7DKangaroo-DriftingDeadline-V1%7B%5C_%7D2-User%7B%5C_%7DGuide.pdf.

[Gho+16] Debopriya Ghosh et al. "Big Data-based Smart City Platform". In: Proceedings of the 17th Internation New York, New York, USA: ACM Press, 2016, pp. 58–66. ISBN: 9781450343398. DOI: 10.1145/2912160.2912205. URL: http://dl.acm.org/citation.cfm?doid=2912160.2912205.

[Teh16] N. Tehrani. "Extraversion, neuroticism and secondary trauma in Internet child abuse investigators". In: Occupational Medicine 66.5 (July 2016), pp. 403–407. ISSN: 0962-7480. DOI: 10.1093/occmed/kqw004. URL: https://academic.oup.com/occmed/article-lookup/doi/10.1093/occmed/kqw004.

[AB17] Emilio Ancillotti and Raffaele Bruno. "Comparison of CoAP and CoCoA+ congestion control mechanisms for different IoT application scenarios". In: 2017 IEEE Symposium on Computer IEEE, July 2017, pp. 1186–1192. ISBN: 978-1-5386-1629-1. DOI: 10.1109/ISCC.2017.8024686. URL: http://ieeexplore.ieee.org/document/8024686/.

[BAA17] Majdi Beseiso, Abdulkareem Al-Alwani, and Abdullah Altameem. "An Interoperable Data Framework to Manipulate the Smart City Data using Semantic Technologies". In: International Journal of Advanced Computer Science and Applications 8.1 (2017). ISSN: 21565570. DOI: 10.14569/IJACSA.2017.080110. URL: http://thesai.org/Publications/ViewPaper?Volume=8%7B%5C&%7DIssue=1%7B%5C&%7DCode=ijacsa%7B%5C&%7DSerialNo=10.

[FDA17] FDA. Cybersecurity Vulnerabilities Identified in St. Jude Medical's Implantable Cardiac Devices and 2017. URL: https://www.fda.gov/medical-devices/safety-communications/cybersecurity-vulnerabilities-identified-st-jude-medicals-implantable-cardiac-devices-and-merlinhome.

[HV17] Malek Harbawi and Asaf Varol. "An improved digital evidence acquisition model for the Internet of Things forensic I: A theoretical framework". In: 2017 5th International Symposium IEEE, Apr. 2017, pp. 1–6. ISBN: 978-1-5090-5835-8. DOI: 10.1109/ISDFS.2017.7916508. URL: http://ieeexplore.ieee.org/document/7916508/.

[Jun+17] Adelson Araujo Junior et al. “A predictive policing application to support patrol planning in smart cities”. In: 2017 International Smart Cities Conference (ISC2). IEEE, Sept. 2017, pp. 1–6. ISBN: 978-1-5386-2524-8. DOI: 10.1109/ISC2.2017.8090817. URL: http://ieeexplore.ieee.org/document/8090817/.

[Lyo+17] Kate Lyons et al. “A Profile of Injuries Sustained by Law Enforcement Officers: A Critical Review”. In: International Journal of Environmental Research and Public Health 14.2 (Feb. 2017), p. 142. ISSN: 1660-4601. DOI: 10.3390/ijerph14020142. URL: http://www.mdpi.com/1660-4601/14/2/142.

[MK17] Lars Marcus and Daniel Koch. “Cities as implements or facilities – The need for a spatial morphology in smart city systems”. In: Environment and Planning B: Urban Analytics and Cit 44.2 (Mar. 2017), pp. 204–226. ISSN: 2399-8083. DOI: 10.1177/0265813516685565. URL: http://journals.sagepub.com/doi/10.1177/0265813516685565.

[Mef+17] Christopher Meffert et al. “Forensic State Acquisition from Internet of Things (FSAIoT)”. In: Proceedings of the 12th International Conference on Availability, Reliability and Secu New York, NY, USA: ACM, Aug. 2017, pp. 1–11. ISBN: 9781450352574. DOI: 10.1145/3098954.3104053. URL: https://dl.acm.org/doi/10.1145/3098954.3104053.

[NB17] Raja Jitendra Nayaka and Rajashekhar C. Biradar. “Data aggregation and routing scheme for smart city public utility services using WSN”. In: 2017 Second International Conference o IEEE, Feb. 2017, pp. 1–8. ISBN: 978-1-5090-3239-6. DOI: 10.1109/ICECCT.2017.8117949. URL: http://ieeexplore.ieee.org/document/8117949/.

[Pri17] Marilyn Price. “Psychiatric Disability in Law Enforcement Officers”. In: Behavioral Sciences & the Law 35.2 (Mar. 2017), pp. 113–123. ISSN: 07353936. DOI: 10.1002/bsl.2278. URL: http://doi.wiley.com/10.1002/bsl.2278.

[Sha+17] Abhay Sharma et al. “Schemas for IoT interoperability for smart cities”. In: Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Enviro New York, New York, USA: ACM Press, 2017, pp. 1–2. ISBN: 9781450355445. DOI: 10.1145/3137133.3141466. URL: http://dl.acm.org/citation.cfm?doid=3137133.3141466.

[VUL17] VULDB. FFMPEG UP TO 3.3.4 LIBAVCODEC/FFV1DEC.C READ_HEADER MP4 FILE MEMORY COR 2017. URL: https://vuldb.com/?id.109163.

[Bab+18] Leonardo Babun et al. “IoTDots: A Digital Forensics Framework for Smart Environments”. In: (Sept. 2018). arXiv: 1809.00745. URL: http://arxiv.org/abs/1809.00745.

[Dal+18] Janis Dalins et al. “Laying foundations for effective machine learning in law enforcement. Majura – A labelling schema for child exploitation materials”. In: Digital Investigation 26 (Sept. 2018), pp. 40–54. ISSN: 17422876. DOI: 10.

1016/j.diin.2018.05.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S1742287618301555.

[Jar+18] Yaser Jararweh et al. "An experimental framework for future smart cities using data fusion and software defined systems: The case of environmental monitoring for smart healthcare". In: Future Generation Computer Systems (Mar. 2018). ISSN: 0167739X. DOI: 10.1016/j.future.2018.01.038. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17312013.

[KCK18] George Kokkonis, Angelos Chatzimparmpas, and Sotirios Kontogiannis. "Middleware IoT protocols performance evaluation for carrying out clustered data". In: 2018 South-Eastern Europ IEEE, Sept. 2018, pp. 1–5. ISBN: 978-618-83314-1-9. DOI: 10.23919/SEEDA-CECNSM.2018.8544929. URL: https://ieeexplore.ieee.org/document/8544929/.

[MBS18] Aine MacDermott, Thar Baker, and Qi Shi. "Iot Forensics: Challenges for the Ioa Era". In: 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTM IEEE, Feb. 2018, pp. 1–5. ISBN: 978-1-5386-3662-6. DOI: 10.1109/NTMS.2018.8328748. URL: http://ieeexplore.ieee.org/document/8328748/.

[MBB18] Federico Montori, Luca Bedogni, and Luciano Bononi. "A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring". In: IEEE Internet of Things Journal 5.2 (2018), pp. 592–605. ISSN: 23274662. DOI: 10.1109/JIOT.2017.2720855.

[Net18] Networkb. Golang Fuzzing: A go-fuzz Tutorial and Example. 2018. URL: http://networkbit.ch/golang-fuzzing/.

[Pra+18] Manas Pradhan et al. "Toward an Architecture and Data Model to Enable Interoperability between Federated Mission Networks and IoT-Enabled Smart City Environments". In: IEEE Communications Magazine 56.10 (Oct. 2018), pp. 163–169. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1800305. URL: https://ieeexplore.ieee.org/document/8493137/.

[Sei18] Kathryn C. Seigfried-Spellar. "Assessing the Psychological Well-being and Coping Mechanisms of Law Enforcement Investigators vs. Digital Forensic Examiners of Child Pornography Investigations". In: Journal of Police and Criminal Psychology 33.3 (Sept. 2018), pp. 215–226. ISSN: 0882-0783. DOI: 10.1007/s11896-017-9248-7. URL: http://link.springer.com/10.1007/s11896-017-9248-7.

[Tie+18] Hope M. Tiesman et al. "Nonfatal Injuries to Law Enforcement Officers: A Rise in Assaults". In: American Journal of Preventive Medicine 54.4 (Apr. 2018), pp. 503–509. ISSN: 07493797. DOI: 10.1016/j.amepre.2017.12.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S074937971730716X.

[YHJ18] Liu Yingran, Sun Hailing, and Gong Jian. "Geologic hazard susceptibility and disaster risk mapping based on information value model for the MianChi county, China". In: IOP Conference Series: Earth and Environmental Science 199.2 (2018). ISSN: 17551315. DOI: 10.1088/1755-1315/199/2/022039.

[Ana+19] Ganesh Ananthanarayanan et al. "Video Analytics - Killer App for Edge Computing". In: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and New York, NY, USA: ACM, June 2019, pp. 695–696. ISBN: 9781450366618. DOI: 10.1145/3307334.3328589. URL: https://dl.acm.org/doi/10.1145/3307334.3328589.

[Che+19] Ruilong Chen et al. "Wildlife surveillance using deep learning methods". In: Ecology and Evolution 9.17 (Sept. 2019), pp. 9453–9466. ISSN: 2045-7758. DOI: 10.1002/ece3.5410. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.5410.

[Dig19] Digiteum. Internet of Things for Smart Warehouses. 2019. URL: https://www.digiteum.com/internet-of-things-for-smart-warehouses.

[Din+19] Wenxiu Ding et al. "A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion". In: Information Fusion 51 (Nov. 2019), pp. 129–144. ISSN: 15662535. DOI: 10.1016/j.inffus.2018.12.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1566253518304731.

[ELD19] Hendrik Engelbrecht, Stephan G. Lukosch, and Dragos Datcu. "Evaluating the Impact of Technology Assisted Hotspot Policing on Situational Awareness and Task-Load". In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technolog 3.1 (Mar. 2019), pp. 1–18. ISSN: 24749567. DOI: 10.1145/3314396. URL: http://dl.acm.org/citation.cfm?doid=3323054.3314396.

[Gal19] Maša Galič. "Surveillance, Privacy and Public Space in the Stratumseind Living Lab: The Smart City Debate, beyond Data". In: Tilburg Law School Legal Studies Research Paper Ser (2019).

[IK19] Collins Ineneji and Mehmet Kusaf. "Hybrid weapon detection algorithm, using material test and fuzzy logic system". In: Computers & Electrical Engineering 78 (Sept. 2019), pp. 437–448. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2019.08.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045790619303933.

[Jan19] Subong Jang. "A Study on Internet of Things (IoT) Forensic". Masters thesis. Hallym University, 2019. URL: https://lifs.hallym.ac.kr/pubs/2019-Thesis-MSc-Subong-A%7B%5C_%7DStudy%7B%5C_%7Don%7B%5C_%7DInternet%7B%5C_%7Dof%7B%5C_%7DThings%7B%5C_%7DForensics.pdf.

[Jen19] Ginger Jenkins, Lee. “Negative Appraisal Correlation to PTSD Symptoms Among Law Enforcement Officers”. Doctoral Dissertation. Walden University, 2019, pp. 1–161.

[Kol+19] Sefki Kolozali et al. “Observing the Pulse of a City: A Smart City Framework for Real-Time Discovery, Federation, and Aggregation of Data Streams”. In: IEEE Internet of Things Journal 6.2 (Apr. 2019), pp. 2651–2668. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2872606. URL: https://ieeexplore.ieee.org/document/8476168/.

[Lau+19] Billy Pik Lik Lau et al. “A survey of data fusion in smart city applications”. In: Information Fusion 52 (Dec. 2019), pp. 357–374. ISSN: 15662535. DOI: 10.1016/j.inffus.2019.05.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S1566253519300326.

[MB19] Mila Mileva and A. Mike Burton. “Face search in CCTV surveillance”. In: Cognitive Research: Principl 4.1 (Dec. 2019), p. 37. ISSN: 2365-7464. DOI: 10.1186/s41235-019-0193-0. URL: https://cognitiveresearchjournal.springeropen.com/articles/10.1186/s41235-019-0193-0.

[OR19] Liav Orgad and Wessel Reijers. “A Dystopian Future? The Rise of Social Credit Systems”. In: SSRN Electronic Journal (2019). ISSN: 1556-5068. DOI: 10.2139/ssrn.3491179. URL: https://www.ssrn.com/abstract=3491179.

[Piz+19] Eric L. Piza et al. “CCTV surveillance for crime prevention”. In: Criminology & Public Policy 18.1 (2019), pp. 135–159. DOI: 10.1111/1745-9133.12419. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1745-9133.12419. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1745-9133.12419.

[PMY19] Sandy Suryo Prayogo, Yulisdin Mukhlis, and Bayu Kumoro Yakti. “The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266”. In: 2019 Fourth International Conference on Informatics and Computin IEEE, Oct. 2019, pp. 1–5. ISBN: 978-1-7281-2207-6. DOI: 10.1109/ICIC47613.2019.8985850. URL: https://ieeexplore.ieee.org/document/8985850/.

[SC19] Francesco Servida and Eoghan Casey. “IoT forensic challenges and opportunities for digital traces”. In: Digital Investigation 28 (Apr. 2019), S22–S29. ISSN: 17422876. DOI: 10.1016/j.diin.2019.01.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S1742287619300222.

[SK19] Chanwit Suwannapong and Chatchai Khunboa. “Congestion control in CoAP observe group communication”. In: Sensors (Switzerland) 19.15 (2019). ISSN: 14248220. DOI: 10.3390/s19153433.

[TS19] Jan Terpstra and Renze Salet. “Change and continuity in the police: Introduction to the Special Issue”. In: International Journal of Police Science & Management 21.4 (Dec. 2019), pp. 193–195. ISSN: 1461-3557. DOI: 10.1177/1461355719889474. URL: http://journals.sagepub.com/doi/10.1177/1461355719889474.

[ZCB19] Xiaolu Zhang, Kim-Kwang Raymond Choo, and Nicole Lang Beebe. “How Do I Share My IoT Forensic Experience With the Broader Community? An Automated Knowledge Sharing IoT Forensic Platform”. In: IEEE Internet of Things Journal 6.4 (Aug. 2019), pp. 6850–6861. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2912118. URL: https://ieeexplore.ieee.org/document/8694855/.

[BTS20] Sivadi Balakrishna, M. Thirumaran, and Vijender Kumar Solanki. “IoT Sensor Data Integration in Healthcare using Semantics and Machine Learning Approaches”. In: 2020, pp. 275–300. DOI: 10.1007/978-3-030-23983-1_11. URL: http://link.springer.com/10.1007/978-3-030-23983-1%7B%5C_%7D11.

[CWE20a] CWE-306. CWE-306: Missing Authentication for Critical Function. 2020.

[CWE20b] CWE-307. CWE-307: Improper Restriction of Excessive Authentication Attempts. 2020. URL: https://cwe.mitre.org/data/definitions/307.html.

[CWE20c] CWE-798. CWE-798: Use of Hard-coded Credentials. 2020. URL: https://cwe.mitre.org/data/definitions/798.html.

[CWE20d] CWE-862. CWE-862: Missing Authorization. 2020. URL: https://cwe.mitre.org/data/definitions/862.html.

[CWE20e] CWE-863. CWE-863: Incorrect Authorization. 2020. URL: https://cwe.mitre.org/data/definitions/863.html.

[Jun+20] Daekyo Jung et al. “Conceptual Framework of an Intelligent Decision Support System for Smart City Disaster Management”. In: Applied Sciences 10.2 (Jan. 2020), p. 666. ISSN: 2076-3417. DOI: 10.3390/app10020666. URL: https://www.mdpi.com/2076-3417/10/2/666.

[Sel20] Mats Seljeseth. “matsse/SmartPolice-Interface”. 2020. URL: https://github.com/matsse/SmartPolice-Interface/.

[Sma20] Samsung SmartThings. SmartThings Classic Documentation: Zigbee Primer. 2020. URL: https://docs.smartthings.com/en/latest/device-type-developers-guide/zigbee-primer.html%7B%5C#%7Dread-and-write-attributes.

[YSK20] Muhammad Mudassar Yamin, Andrii Shalaginov, and Basel Katt. “Smart Policing for a Smart World Opportunities, Challenges and Way Forward”. In: 2020, pp. 532–549. DOI: 10.1007/978-3-030-39445-5_39. URL: http://link.springer.com/10.1007/978-3-030-39445-5%7B%5C_%7D39.

[Bor+] M.S. Borella et al. “Internet packet loss: measurement and implications for end-to-end QoS”. In: Proceedings of the 1998 ICPP Workshop on Architectural and OS Support for Mult IEEE Comput. Soc, pp. 3–12. ISBN: 0-8186-8657-X. DOI: 10.1109/ICPPW.1998.721868. URL: http://ieeexplore.ieee.org/document/721868/.

Mats Didriksen Seljeseth

NTNU
Kunnskap for en bedre verden