A cloud based business intelligence framework for a cellular Internet of Things network

LW Moolman

Orcid.org/0000-0002-2991-4450

Dissertation accepted in fulfilment of the requirements for the degree Master of Engineering in Computer and Electronic Engineering at the North-West University

Supervisor: Prof JEW Holm

Graduation: May 2020

Student number: 24075477

Acknowledgements

I would like to express my sincere appreciation to the following persons involved in the successful completion of my Masters dissertation:

> To my supervisor, Prof. J.E.W. Holm, thank you for all the support, guidance and motivation throughout this entire process. You are truly a great inspiration and role model.
>
> To Rossouw van der Merwe, Pieter Jordaan, Nicojan Vermaak and the entire team at Jericho Systems, thank you for all the emotional and financial support that allowed this dissertation to be completed.
>
> To my parents, Leonie and Gert and brother Jacques Moolman, thank you for all the prayers, words of encouragement and support throughout all of my studies.
>
> To Suanne Bosch, thank you for the encouragement in times of doubt and for all the love and support through all of the late nights required to complete this dissertation.

And finally I would like to thank God, for all the strength and determination He has given me to overcome all of the challenges I have faced.

Abstract

In this research, a Business Intelligence (BI) framework for a cellular Internet of Things (IoT) environment is researched, designed, implemented and evaluated. The BI framework provides a structure that supports development of a BI platform (solution) by first defining a structured platform to provide data, and then following a process flow to ensure valid Artificial Intelligence (AI) models are created. Systems Engineering (SE) principles were applied to define the BI framework, with theoretically grounded Data Mining methods included in the process flow. This system under evaluation is a cellular IoT network of edge devices linked to the cloud via secure, managed data channels. By applying the BI framework, a BI platform is designed and implemented to extract insights from the management data provided by the system. In addition, by following the BI framework's process flow model, AI models are fitted to the available data and included in the BI platform as a total solution.

From the BI platform, insights extracted from data are converted into key performance indicators, or used in models to predict or classify anomalies that indicate operational failures (risk). These models include time series anomaly detection, clustering and classification models.

The research was conducted in a Design Science Research paradigm, with Action Design Research as the method with which to conduct the action research. Quality Research Management was used to provide traceability and to ensure the defined goals were achieved in a systematic manner. Research challenges were identified from observations and a literature survey, researched in literature focus areas, systematically addressed by means of synthesis from literature and creative input, and implemented as a means of validation. The final BI platform solution was applied to real-world data and successfully addressed the initial research challenge.

Keywords: *Business Intelligence, Machine Learning, Internet of Things, Artificial Intelligence, Design Science Research, Data Mining, Systems Engineering*

Opsomming

In hierdie navorsing is 'n raamwerk vir besigheids intelligensie (BI) vir 'n sellulêre Internet van Dinge (IvD) omgewing nagevors, ontwerp, geïmplementeer en geëvalueer. Die BI-raamwerk bied 'n struktuur wat die ontwikkeling van 'n BI platform (oplossing) ondersteun deur eers 'n gestruktureerde platform te definieer wat data verskaf, en volg dan 'n prosesvloei om te verseker dat geldige kunsmatige intelligensie (KI) modelle geskep word. Stelselingenieurswese beginsels is toegepas om die BI-raamwerk te definieer, met teoreties gegronde data ontginningmetodes ingesluit in die prosesvloei. Die huidige stelsel onder ëvaluasie, is 'n sellulêre IvD stelsel van randtoestelle gekoppel aan die wolk via veilige, bestuurde data kanale. 'n BI platform is ontwerp en geïmplementeer deur van die BI-raamwerk gebruik te maak om insigte te ontgin uit die bestuursdata voorsien deur die stelsel. Deur die BI-raamwerk se prosesvloeimodel te volg word KI-modelle op die beskikbare data gepas en word die modelle dan by die BI-platform ingesluit as 'n totale oplossing.

Die BI-platform word gebruik om insigte te ontgin, hierdie insigte word dan omskakel na sleutelprestasie-aanwysers, of in modelle gebruik. Hierdie modelle word dan gebruik om anomalieë te voorspel of te klassifiseer wat dui op operasionele foute (risiko). Hierdie modelle sluit anomalie opsporing in tydreeks data, data groepering en klassifikasie in.

Die navorsing is uitgevoer in 'n ontwerp wetenskaplike navorsing ("Design Science Research") paradigma, met aksie ontwerp navorsing ("Action Design Research") as die metode wat gebruik is om die navorsing me uit te voor. Kwaliteitsnavorsingsbestuur is gebruik om deursigtigheid te verseker terwyl die gedefinieerde doelstellings bereik word op 'n sistematiese wyse. Navorsingsuitdagings is geïdentifiseer vanaf observasies en 'n literatuurstudie, opgebreuk in literatuurfokusareas, is stelselmatig aangespreek deur middel van sintese uit literatuur en kreatiewe insette. Die finale BI-platform is toegepas in 'n regte-wêreld probleem en spreuk die aanvanklike navorsingsuitdaging suksesvol aan.

Sleutelwoorde: *Besigheids Intelligensie, Masjienleer, Stelsels Ingenieurswese, Ontwerp Wetenskaplike Navorsing, Data Ontginning, Internet van Dinge*

Contents

List of Figures

List of Tables

List of Abbreviations

ADR	Action Design Research
AI	Artificial Intelligence
ANN	Artificial Neural Network
ANP	Active Network Provider
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
AUC	Area Under the Curve
BI	Business Intelligence
CRISP-DM	Cross-Industry Standard Process for Data Mining
DSR	Design Science Research
ETL	Extract Transform Load
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GSM	Global System for Mobile communication
IoT	Internet of Things
IQR	Interquartile Range
KB	Knowledge Base
KDD	Knowledge Discovery from Data
KPI	Key Performance Indicator

LSTM	Long Short-Term Memory
MA	Moving Average
ML	Machine Learning
OLAP	Online Analytical Processing
PRC	Precision-Recall Curve
QRM	Quality Research Management
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
ROI	Return on Investment
SaaS	Software as a Service
SARIMA	Seasonal Auto Regressive Integrated Moving Average
SE	Systems Engineering
SoS	System of Systems
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPM	Technical Performance Measure
TPR	True Positive Rate

Chapter 1

Introduction

Cellular networks used in Internet of Things (IoT) applications are often ill-characterised and the users of such networks are often subjected to an environment over which they have very limited control. In addition, cellular modems (or edge devices) are not always informative on their health status as the market is quite competitive and costs are saved by reducing functionality, of which health status is one such less important function. Also, managed networks are not often used due to cost constraints, but provide valuable information for Business Intelligence (BI) purposes. The real world need in this research is to assist a client in the process of establishing a BI platform for a cellular IoT network. The client should be able to follow a process in the future that will result in additions to the BI platform without having to repeat the work done in this study. As a result, a BI framework is required in the form of a process flow model (that is, a general process) that may be used to address this need. By following this process and all the guidelines associated with the process, a BI platform must be provided to run on the client's existing cloud services platform.

The purpose of this study is thus to synthesise and evaluate a BI framework for a cellular IoT environment. This is achieved by conducting research in a Design Science Research (DSR) paradigm to solve the real world problem above, which is in short, to implement a BI platform (solution) for an existing cellular IoT network. This research follows a Quality Research Management (QRM) process [6] that includes extraction of research challenges, design and evaluation of a solution, and instantiating an artefact in the form of a

BI process flow model and BI platform (resulting from the process flow), and also generating knowledge to add into the existing Knowledge Base (KB). An Action Design Research (ADR) method is followed as this research is being conducted while a system is being designed, implemented and evaluated.

1.1 Overview

This research is divided into six separate chapters with the first providing an introduction.

- Chapter 2 - This chapter presents the research methodology used to conduct this research, including Design Science Research (DSR), Action Design Research (ADR),and Quality Research Management (QRM). The chapter describes the DSR paradigm and how it can be used to solve a real world problem and by using ADR to add new knowledge into the KB in the from of artefacts and meta-artefacts. A description of QRM is provided to ensure visibility is provided on how the different aspects of the research are verified and validated;
- Chapter 3 - This chapter consists of extracting, defining and verifying research challenges from a real world problem. The main research challenge is defined as the lack of an integrated BI framework focused on a cellular IoT environment;
- Chapter 4 - This chapter contains a literature review on the literature focus areas required to verify the research challenges and to validate the proposed research solutions. An overview of the architecture and components included in a IoT system is described providing insight into the different layers of an IoT system. Cellular communication systems are researched to provide understanding of the fundamental network characteristics that determine effectiveness. BI is defined and key aspects and challenges are discussed. An overview of available BI frameworks and process is provided. Data mining was researched, focusing on the Cross-Industry Standard Process for Data Mining (CRISP-DM) process and Machine Learning (ML) techniques suitable for anomaly detection on different data types and formats. An overview of Systems Engineering (SE) is provided for a definition of a system and a System of Systems (SoS). The SE process and the importance of a full life-cycle approach to implementing a system are further discussed;

- Chapter 5 - This chapter consists of the synthesis of a BI framework. This framework addresses the need for a general IoT framework with which to develop BI platforms for cellular IoT networks. The BI framework comprises two phases, namely (i) a Development phase, and (ii) an Operations phase. A solution is implemented applying the Development phase of the BI framework, resulting in a platform that enabled insight extraction from the data sources available to the system. The Operational phase process flow model of the BI framework is executed using the implemented platform by running a series of experiments, which provided different anomaly detection models including time series anomaly detection, clustering and classification models;
- Chapter 6 - This chapter summarizes the research challenges and corresponding solutions, and shows how the artefacts produced from this research validate the research challenges and solutions. Traceability is shown using a validation matrix that indicates the contributions of the different literature information sources, literature focus areas, and specific solutions to the research challenges and research solutions.

Chapter 2

Research Methodology

2.1 Design Science Research

The research conducted in this dissertation is directed towards providing a solution to a real-world problem and is best conducted in a Design Science Research (DSR) paradigm [7], which is a problem-solving paradigm [7] suitable for directed research. DSR comprises three primary cycles, as shown in Figure 2.1 below [1]:

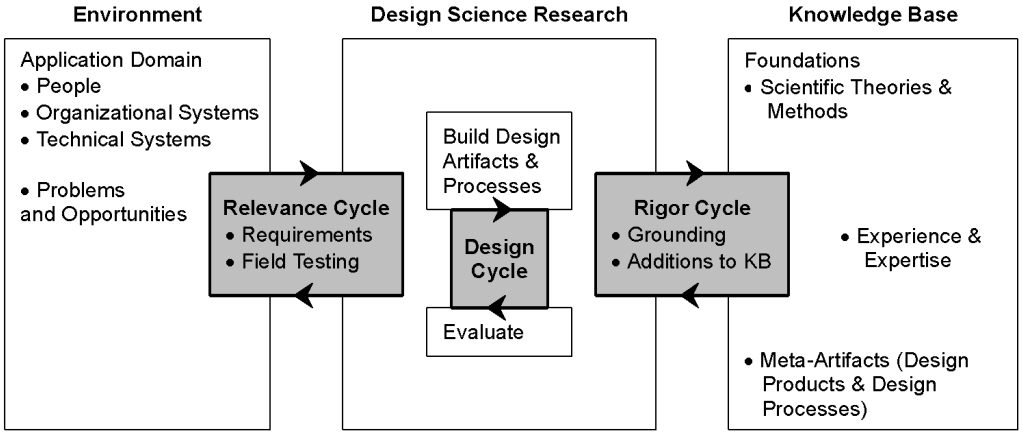

Figure 2.1: The Design Science Research cycles [1]

The Relevance Cycle converts real-world needs and requirements for consideration in the DSR project, and upon completion, verifies and validates the designed solution against these requirements to confirm compliance. The Design Cycle balances real-world requirements with solutions extracted from

the KB, as well as against creative input from the research effort, where the Rigor Cycle is used to ensure grounded theory is applied in creating such a solution. The focus is on providing an artefact, with knowledge added to the KB in the process. The provision of an artefact is key to the DSR process [7] [8] [9] [1] [2]. In this research, the artefact is a BI framework, applied to guide the creation of BI for a cellular IoT network. Requirements were derived from a real-world environment, where units in the field communicate through a managed network to a cloud, as well as with other devices connected through the cellular network. The KB, in this research, is the set of well-researched methods in Artificial Intelligence (AI), as well as experience and knowledge from experts in the field of AI. In this case, knowledge will be added to the KB as part of this research, which is a characteristics of Action Design Research [3], which will be described in the following section. The contribution from this research is thus to provide a framework for BI in cellular IoT systems. This is not a new concept or an invention, but rather a new solution (in its integrated form) to an existing problem (refer to Figure 2.2). As from Hevner [2], no design or research is really "new" as all solutions build on previous concepts and ideas. Therefore, this research is positioned as an improvement in the contribution framework [2].

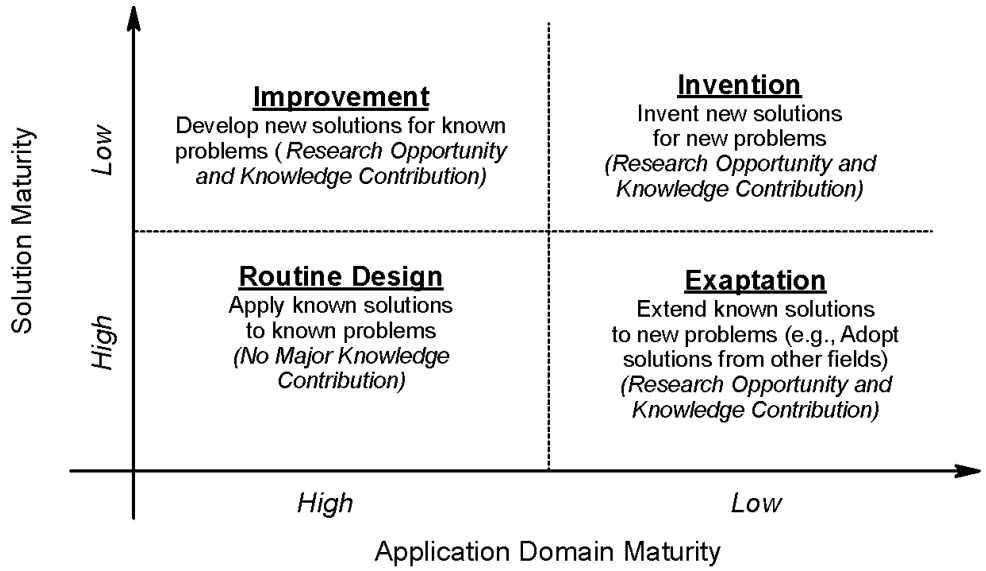

Figure 2.2: DSR knowledge contribution framework [2]

2.2 Action Design Research

ADR is an outflow of Action Research and DSR, with the focus on designing as opposed to simply conducting pure scientific (cause-effect) research [10]. ADR has 4 stages and adheres to 7 principles, as follows [3]:

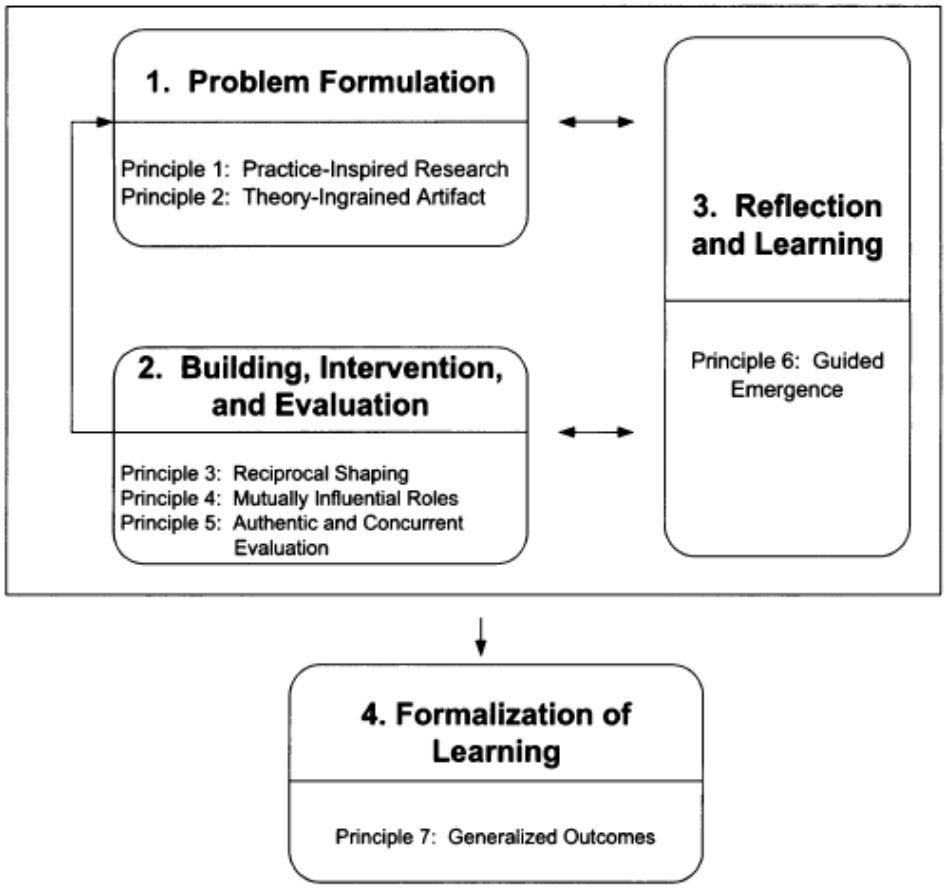

Figure 2.3: ADR stages and principles [3]

For each design stage, specific principles apply, described as follows:

- Stage 1: Problem Formulation
  - Principle 1 – Practice-Inspired Research: This principle aligns with the DSR paradigm in that the research must solve a problem relevant to the real world, i.e. a practical problem. The focus is not primarily on knowledge creation, but to conduct research that produces both solutions to real-world challenges and knowledge that describes and supports solution of a class of similar problems.

  - Principle 2 – Theory-Ingrained Artefact: It is critical that the solution created by research be based on sound theoretical principles. That is, theory applied to the design of an artefact must be of grounded nature. This implies that the designed artefact, although it may be based on prior designs, may be derived or constructed from theory (including functional analyses, for example) that has been proven valid. Theory may be used to structure a problem (analyses and statement), identify solutions election and evaluation), and guide design (constraints and goals).

- Stage 2: Building, Intervention and Evaluation
  - Principle 3 – Reciprocal Shaping: It is almost always the case that a design process comprises a number of iterations before the final design emerges. There is constant interaction between the real world and the abstracted world as new perspectives are formed during the analysis and design phases, and the test and evaluation phases in the real world. The design is thus shaped by the real world, and the real world may change according to the design in a reciprocal way.
  - Principle 4 – Mutually Influential Roles: This principle is based on the different roles played by action design researchers and the real world practitioners.The information, experience and creative shared by both paradigms hold mutual benefits for both real world and theoretical world. Roles are often shared, where a researcher may be active in practice, and a practitioner may conduct research.
  - Principle 5 – Authentic and Concurrent Evaluation: An iterative approach to design includes the process of ongoing evaluation, and the process of design and evaluation is effectively merged. That is to say, the design is constantly evaluated and results used to affect change in the design, and so on. Thus, the design is strongly influenced by the real world since requirements from the real world are used to evaluate the artefact.
- Stage 3: Reflection and Learning
  - Principle 6 – Guided Emergence: Design is a deliberate act of creating a solution from specific requirements (or goals) in a focused effort. Emergence implies that a design should be formed in an almost organic way, which is contradictory to formal design. However, by allowing freedom in the design process, it is possi-

ble to adapt the design not only to meet set requirements, but also to allow feedback and creative input to achieve the design goals. Guided emergence thus requires both boundaries and goals to form a solution, which is typical to a creative process that uses reflection (i.e. critical evaluation) to influence the design, often in profound ways.

- Stage 4: Formalization of Learning
  - Principle 7 – Generalized Outcomes: This is a critical principle in the ADR process as it is based on a form of abstraction and generalization. Abstraction allows generalization to take place, where generalization is aimed at addressing more than the current real-world problem. In essence, a class of problems may be addressed by a generalized design (and its associated design theories) as opposed to providing a specialized solution.

In this research, the artefact is in the form of a framework that includes a process and method. The fact that a process is provided supports the notion of a generalized design that is aimed at solving a class of problems, namely to provide BI for management of cellular networks. The design is practice inspired as it addresses a real-world challenge, and is based on sound theory of AI (which, in turn, is based on statistics and probability theory, pattern recognition, and time series analyses theories). Reciprocal shaping is a consequence of the interaction between measured data and feature extraction, combined with constant evaluation of the artefact by means of experiments. The researcher, in this case, is also a practitioner that works with cellular IoT networks on a regular basis, hence the presence of mutually influential functions. The application of concurrent evaluation is inherent to AI problems since models are constantly evaluated against practical data by (i) manually extracting features from real-world data, (ii) training models on data sets, (iii) evaluating model performance also on real-world data sets, and (iv) adapting and improving models in an iterative manner. The fact that the final solution (in the form of a framework) is formed by means of emergence in that the model has been reiterated based on reflection (critical evaluation and feedback) throughout the design process.

2.3 Quality Research Management

The research process was managed using Quality Research Management to ensure focus is maintained on the research requirements [6]. The process provides a means to trace research requirements to solutions in the design process, provides visibility of the research process (and requirements), and ensures validation and verification is achieved in a formal way. Matrices are used to capture requirements and to allocate solutions to requirements in a structured way, as presented in the chapters that follow. In this dissertation, research challenges were derived from a real-world case study, literature survey and expert inputs. These challenges are derived in Chapter 3 and are addressed by concept solutions. Literature topics were identified in Chapter 4 to elaborate on, and confirm, the research challenges and concept solutions. The design then focused on creative input, guided emergence, and existing design solutions to provide an integrated framework. Experiments in Chapter 5 allow for critical evaluation of specific solutions and the final integrated framework is then formed based on synthesis from literature, creative input and critical evaluation. The final framework provides a process that can be followed to put a BI solution in place for IoT communications networks.

2.4 Summary

The research conducted in this project is aimed at solving a real-world problem in a DSR paradigm, using principles of ADR and being managed by QRM. The end result will be an artefact in the form of a BI framework for future use in development of BI solutions in practice. This is the main artefact of the research, but is supported by methods that have been evaluated in experiments. These methods are used when applying the framework process and are considered to be grounded theoretical elements of the framework, applied to a real-world problem.

Chapter 3

Problem Statement

This research was conducted inside the DSR paradigm by following an ADR methodology, managed using QRM. This methodology consists of evaluating a real world problem, extracting research challenges that define the shortfalls to be addressed, defining concept solutions, and then providing detailed solutions to each of the concept solutions. The real world problem evaluated in this dissertation was briefly described in Chapter 1, but is analysed here to provide more clarity on the actual problem.

A cloud based BI platform is required to improve the visibility of performance metrics and to address failures (risks) associated with a cellular IoT network. This system is described in more detail in Section 5.2.

DSR uses a relevance cycle to evaluate information sources and extract research challenges from these sources. The sources and research challenges are described below and how these sources validate the challenges.

3.1 Information Sources

3.1.1 Real world problem and need

The current cellular IoT network provides a communication link between client application services and edge devices. This system includes a cloud-based maintenance component that generates and stores maintenance data. There is a real world need to extract insights from this data to improve performance and reduce operational risk. The information to validate this

challenge (i.e the information source) is an expert on the client's network, and observation of the client's system architecture and resources confirmed this need. The need thus exists for an integrated BI platform / solution. Furthermore, the client requires a process to follow for future expansion of the system in case more data becomes available, hence the need exists for a process flow model that can address the need for future expansion and improvement.

3.1.2 BI Publications

Many BI publications and sources provide a description of the architecture and implementation of a BI solution, or the processes involved in knowledge discovery. A need exists to incorporate the implementation and operation of a BI solution based on a systems engineering full life-cycle approach (to allow for future iterations, upgrades, or platform changes). The literature sources also indicated a lack of a overall ontology for BI systems specifically in a cellular IoT environment.

3.1.3 Cellular network observation

In order to improve the performance of a network, it is important to identify and define the core characteristics of the system. By evaluating the current client system, it was observed that the system characteristics used to define and understand the performance of the system were largely undefined. This also indicated that a integrated application is required to continuously extract insights from current system data in an effort to improve system performance.

3.2 Research Scope

The research scope is defined from the information sources described above and is presented as research challenges:

- Lack of implementation framework for BI in a cellular IoT network - The challenge is thus the absence of a BI process flow model that can be applied to generate a BI platform as a solution to the cellular IoT network need. The process flow model is a general process the client can follow to produce more BI platforms in the future. Therefore, this research is not just another "standard design"as it abstracts and generalizes the real world problem in the DSR context;
- Unknown system characteristics - A large part of the BI solution is to increase visibility of the performance and failures associated with the IoT system. To achieve this, the system characteristics represented by measurements (thus, measurement data) need to be analyzed and defined. This can also be considered as the key performance indicators (KPIs) of the IoT network (system) under evaluation;
- Lack of an intelligence ontology - The ontology associated with the BI framework and the IoT system under evaluation are undefined. This can cause uncertainty and misalignment due to different perspectives of system users. The ontology includes terms and concepts used in the BI framework to describe the structure, components and interfaces used in the BI framework. This also includes the definition of key concepts and terminology required to describe the IoT network under evaluation;
- Lack of integrated application - A need for an integrated BI solution consisting of different visualization, alerting and reporting components is required. These components should improve the overall system performance by indicating risks, system performance and enabling insights to be extracted from the system data.

3.3 Summary

The main research challenge is as follows:

Research, synthesize and evaluate an integrated implementation and operational BI framework to run on a BI platform for a cellular IoT network.

Table 3.1 shows a summary of the research challenges and the validation of these challenges from relevant information sources. The research challenges resulting from the high level analysis are shown in the columns of the table, and the sources that defined the challenges are shown in the rows. The arrows associate information sources to challenges and are pointing towards the challenges to show the logical flow from source to challenge.

Table 3.1: Research challenges summary

		Research Challenges			
		Lack of implementation framework for BI in cellular IoT	System characteristics unknown	Lack of intelligence ontology	Lack of integrated application
Information Sources	Real world definition of need	↑			↑
	Observations from Cellular Network applications		↑		↑
	BI Publications	↑		↑	

Each research challenge above will be further investigated in the literature study in Chapter 4 by studying literature relevant to the challenge. Each challenge will then be addressed by a concept solution, which will in turn be addressed by specific solutions in Chapter 5. By linking challenges to solutions, traceability is introduced and the reader can follow the logical flow from challenge to solution in a systematic manner.

Chapter 4

Literature Study

In this chapter a literature study is conducted on IoT, BI, data mining, cellular communication systems and Systems Engineering (SE). In order to contribute to the research challenges presented in 3 a literature review is required on the above mentioned research fields. Research on BI is required to understand the components included in a BI solution, to evaluate existing frameworks, to identify challenges and pitfalls in existing implementations and to define the ontology associated with a BI solution. It is required to understand an IoT system and how BI can be used to add value to an IoT system. In order to implement and evaluate intelligent models in a BI system, an understanding of data mining and Machine Learning (ML) is required. Background on cellular communication systems is required to fully understand and define the dynamics and system characteristics that describe the performance of the system under evaluation. Finally to propose a new BI framework, SE concepts and system life-cycles needs to be evaluated.

4.1 IoT

Gartner defines IoT as a network of interconnected devices or things that can collect data or sense internal or external states and interact with the environment. IoT also includes connecting assets, processes and personnel to allow improving business processes using data collated by these devices [11].

IoT has become very relevant and wide spread in many economic sectors, with implementations in medical, smart cities, automated industries, smart agriculture, security and many more [12][13].

With the adoption of IoT in many businesses, the amount of available data in a business increases and this creates a need to effectively evaluate and process this data into insights. It is thus important to understand the structure of an IoT system and how data interacts with the different components and business users contained in an IoT system.

4.1.1 IoT in general

The IoT is a network of interconnected sensors, control units, users and applications that is enabled by an ecosystem comprising different elements [14], as follows:

- Hardware elements: All distributed sensors, control systems, communication devices and possibly server infrastructure that are interconnected by means of the internet;
- Interconnection networks: The network system that supports interconnectivity in the form of distributed network infrastructure (LoRa, Sigfox, cellular networks, and others) as well as larger backhaul networks higher up in the hierarchy;
- Remote access: Applications and supporting infrastructure that provides users with access to operational data, decision support information and other BI;
- Platform / infrastructure: Typically, software and hardware in the cloud that hosts the messaging, analytics and storage of IoT solutions;
- Security: All aspects of security (physical and cyber) that secure an IoT solution's data and applications both in the cloud, on the edge, and in the network.

It is clear that the interconnection network forms the backbone of an IoT solution and availability of connectivity is a critical aspect in this regard. The focus in this research is thus on ensuring the availability of the IoT network. This is done by providing a framework that provides communication characteristics.

Different networks are used to provide interconnectivity, of which Wi-Fi, cellular, mesh, and low power networks are mostly used. Of these, cellular networks are the focus of this research as South Africa does not have the fibre backhaul infrastructure of a first world country and IoT thus relies heavily on cellular communication.

Worldwide, in 2018, wi-fi networks provided around 80% of connectivity, with cellular networks providing around 60% in the second place. LTE-M is currently advancing as a low power alternative to conventional cellular networks and will most likely be the most attractive choice for IoT in the near future due to its high bandwidth and extensive coverage offered by network providers[14].

The focus in IoT, apart from system elements as discussed above, is on artificial intelligence (AI) and applications that use AI are deployed at an increasing rate. AI also forms part of this research in that it will be applied to extract communication characteristics and anomalies for management purposes.

A general view of a cloud based IoT architecture relevant to this research is shown in Figure 4.1, with typical elements of an IoT network. The extraction of data from the network is shown with management information and BI indicated at the top of the diagram. Through the internet, big data is acquired and analyzed to provide operational control information, management information, and BI. The network of interest is shown to show the scope of this research (all other networks are assumed to connect to the cloud via the cellular network for the purpose of this research). The extraction of information and intelligence forms part of this research in that the network behaviour and status will be estimated (from data) and anomalies be raised using AI methods.

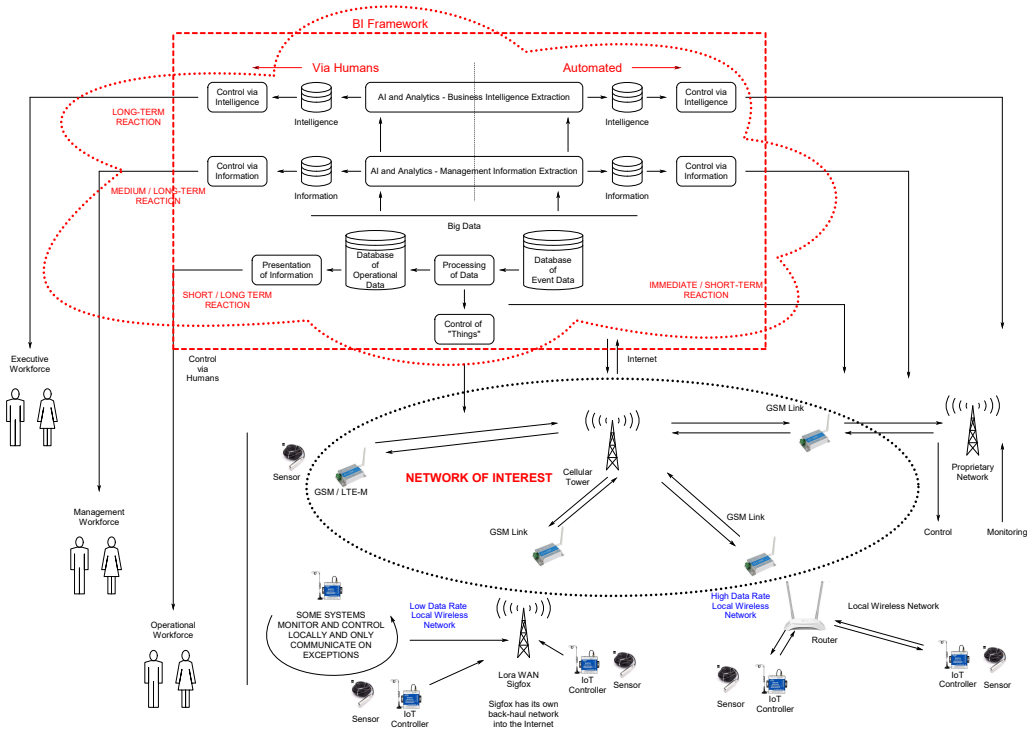

Figure 4.1: IoT Architecture

4.1.2 IoT Layers

Literature sources describe an IoT system as having different layers [15][16][17][18], where these sources differ slightly in their descriptions of the different IoT layers. A basic IoT system will often include at least 4 layers, shown in Figure 4.2 described as follows [16][17][19]:

- Sensing/Control Layer - This contains the edge IoT devices including sensors and actuators connected to the physical world;
- Networking Layer - This layer, also referred to as the communication layer, contains all of the technologies and infrastructure required to connect the IoT devices to the rest of the system, allowing data to be exchanged between the layers;
- Processing Layer - This layer, also referred to as the middleware or service layer, contains the components required to manage and convert the data into services that can be accessed by the interface layer;

- Application Layer - This layer, also referred to as the interface layer, contains the applications and tools that interfaces the service layer and the end user to allow the end users to access the main application of the IoT system.

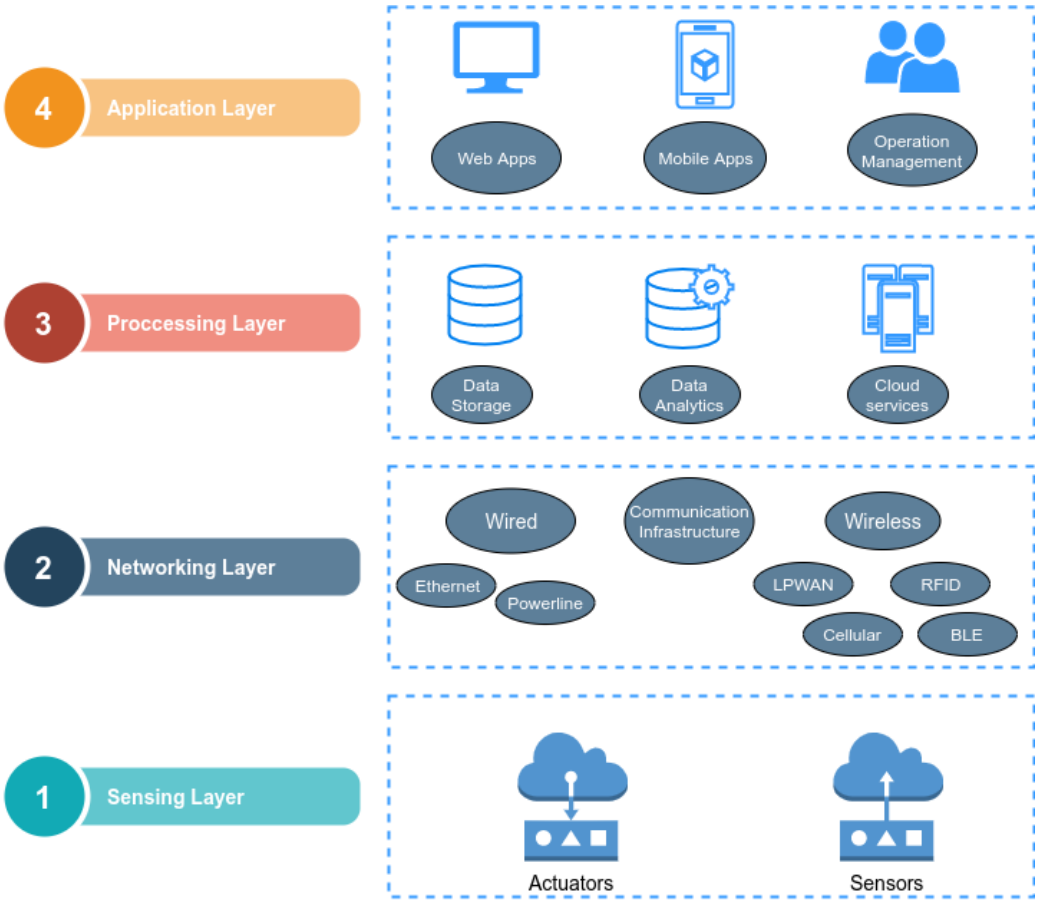

Figure 4.2: Physical IoT Layers

Another approach is to divide the layers based on different types of operators and data produced in the system shown in Figure 4.3.

- Technical Layer - This layer contains the physical edge devices. These can be sensors, actuators or any other IoT devices. This layer contains real time data in large volumes and low density. This is the source data generated by the core IoT devices. The data is used to make immediate decisions relating to the infrastructure or equipment of the system;
- Operations Layer - This layer contains users and processes that form the core operations of the system. The data in this layer is more dense and can be considered as information on the core operations or services. This includes tactical decisions having a short term effect;
- Information Layer - This layer contains system information. Thus the data has been aggregated or analyzed into useful information that allow tactical-strategic decisions affecting the management of the system. This includes managing operational risk and improving core process performance;
- Business Layer - This layer consists of data converted into system intelligence or insights that allow decisions that have a long term effect. This includes strategic-tactical system management focusing on managing enterprise risk and optimising the processes to achieve improved system performance.

It is important to understand that different system operators function in each layer. These operators are required to execute tasks based on the data available. These tasks can be considered as business decisions, where these decisions range from short term decisions based on real time or low density, high volume data to long term decisions based on high density, low volume information or intelligence.

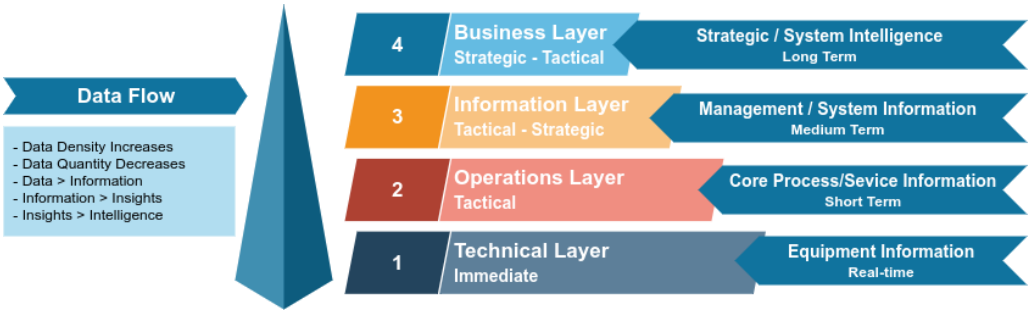

Figure 4.3: Data Focused IoT Layers

4.2 Cellular Communication Systems

Cellular networks provide infrastructure for IoT systems in many applications, including the systems considered in this research. The fundamental network characteristic that determines network effectiveness is its availability, where in this case the cellular network is used to provide data between edge devices, as well as from edge device to the cloud in a reliable manner. Network operators do not provide availability data as part of their service, but the Global System for Mobile communication (GSM) standard defines parameters that are visible to edge devices. In addition, the edge devices provide additional parameters that may be used to characterize the system, such as device battery status and reboot data, amongst others.

GSM networks are characterized by parameters typical to wireless systems, where the fundamental principles are discussed here. A typical environment is shown in Figure 4.4, where transceivers in cellular towers communicate with both stationary and mobile devices. Interference signals, path loss, fading and obstructions cause deterioration in signal-to-noise ratio (or rather, Eb/No) which is the fundamental radio frequency parameter when determining data throughput [20]. In addition to signal-to-noise, channel availability is a fundamental cellular parameter that depends on infrastructure, which is dependent on the network protocol and equipment, cellular planning, environmental characteristics, and user density and behaviour. The end user has limited access to parameters such as received signal strength, bit error rate, and data throughput (not directly provided by the network, as such).

For a specific network protocol, including its physical layer that depends on the generation of network (e.g. 2G vs 4G), the bit error rate is typically determined by a device's signal strength [21]. As edge devices have similar noise bandwidths for given protocols, the signal to noise ratio is well indicated by the signal strength. An increase in signal strength results in less errors being made when symbols are detected, which implies a reduction in bit error rate. The bit error rate, however, is not the only factor that determines data throughput as the data rate is also determined by the network air protocol and network congestion (which varies during the day).

In addition to the data provided by the cellular network, edge devices and cloud software services have the ability to measure data throughput, which is the most relevant parameter for representing link quality. Cumulative and differential data usage also indicate network activity specific to an edge device. By using two Active Network Providers (ANPs), it is possible to increase service availability by selecting the most active / available network – this becomes a higher level network characteristic that may be used to detect active network behavioural patterns and anomalies. Additional edge device behaviour that may be used to characterize the network includes the device's status, specifically the status of power, batteries, and possible device reboots.

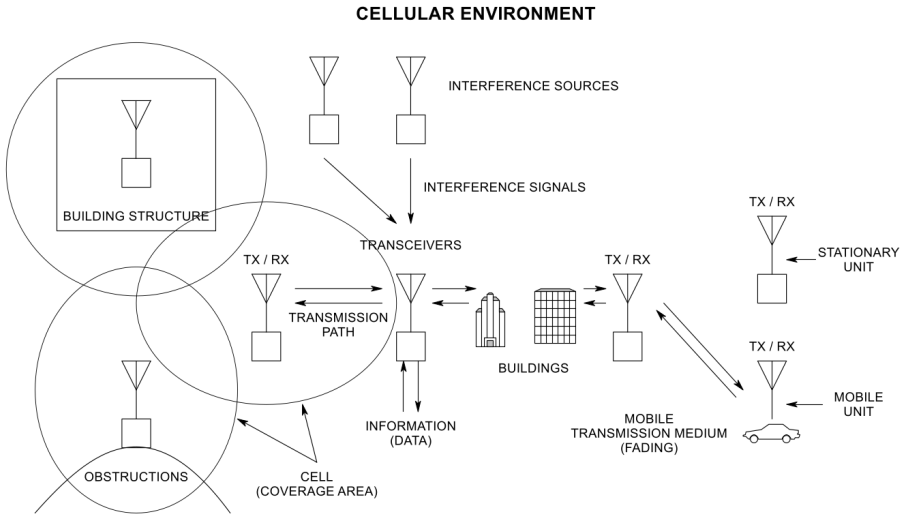

Figure 4.4: Cellular Environment

For this research, a cellular network itself is less important than the overall network system – that is, the network system is broader than just two cellular networks and also includes the edge device and its characteristics. These measurable parameters, together, may be used to estimate the underlying status of the network system and to predict its behaviour. If the behaviour of the network system and the edge devices cannot be predicted, the network system presents an anomaly that must be actioned and resolved in order to restore the communication service.

4.3 Business Intelligence

4.3.1 BI definition

Different definitions for BI can be found, and these definitions vary somewhat. Some define BI as tools and processes while others define BI as an umbrella term containing a wide variety of techniques, methodologies, systems, software, tools etc. [22] . Most definitions agree on the general concept or goal of BI, this being to improve some processes or system using data as the driver to support operational, tactical and strategic business decisions. This is generally achieved by processing data into actionable insights using a variety of analytically techniques combined with business knowledge [23].

A more comprehensive definition relevant for this dissertation is provided in section 5.1.

4.3.1.1 Existing frameworks

BI can be divided into six operational components [24] :

- Source Data - Multiple internal and external raw data sources, which can include unstructured or structured data;
- Extract Transform Load (ETL) - This is the process and tools used to extract data from different sources, format, and clean the data to ensure more reliable information. This can also include aggregating the data into more sensible features or metrics. The data then gets loaded into the Data Warehouse;
- Data Warehouse - A data warehouse describes a collection of all of the data relevant to BI as extracted from internal and external sources. This is usually separated from the operational databases to improve performance and reliability. The data warehouse can be subdivided into data marts containing related data for ease of access and security;
- Online Analytical Processing (OLAP) - OLAP refers to the process of exploring the data using multidimensional cubes to allow comparing and grouping data;
- Visualizations - visualizing data is an important part of BI. This allows different business users to access data and make assumptions based on visual interpretation of the data;

- Dashboards - This provides an overview of the most important information extracted in the BI process. This is usually customized to cater for the specific user.

Liyang *et al.* provides a BI framework based on Software as a Service (SaaS) [25]. This is described as four layers with a fifth layer used to manage all of the layers. These layers consist of :

- Infrastructure Layer : This layer contains the physical components used to host the system. This includes the hardware, software, storage etc;
- Data Service Layer : This layer contains the management and storage of the data used in the system;
- Business Service Layer : This layer consists of four different sub services. These services are Integration Service , Analysis Service, Knowledge Discovery Service and Reporting Services;
- User Interface Service Layer : This layer consists of the components business users use to interface with the BI application;
- Operational Service Layer : This layer is used to manage the other layers with regard to availability, access, scaling, pricing and maintenance.

4.3.1.2 BI Challenges

The following describes some of the main challenges faced when implementing BI [22] [26].

- Bad data quality - When errors in saving or extracting the data occur, the insights gained from this data can be misleading and confusing. This can cause the BI users to distrust the BI system;
- User resistance to BI tools - If the BI tools are not user friendly and relevant to the different business users, the system can easily become a barrier rather than an aid;
- Undefined KPIs resulting in return on investment (ROI) not being measured - It can be very difficult to measure the ROI of a BI implementation and thus it is important to evaluate end explore the important KPIs that can indicate ROI;

- Ineffective business communication - BI should be implemented on different business levels and if the communication between these levels is ineffective or unclear BI opportunities and insights can get lost in translation.

4.4 Data Mining

This section contains an overview of data mining and different sub processes contained in the data mining process. It describes the very popular data mining process called cross-industry standard process for data mining (CRISP-DM). Different supervised and unsupervised Machine Learning (ML) techniques and processes are described. Background on time series forecasting is provided here with the focus on Auto Regressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM) models. An overview of different model evaluation techniques is provided.

4.4.1 Definition

Data Mining, also referred to as knowledge discovery in databases or from data (KDD), can be defined as an interdisciplinary subject that contains different techniques and processes. The process of KDD is defined as the iterative sequence of 7 steps with the goal to extract knowledge from data [27].

The steps are as follows [27]:

- Data cleaning - This consists of removing inconsistent data and noise contained in the data;
- Data integration - This is achieved by combining multiple data sources to create a new source containing the relevant information from the different sources;
- Data selection - Comprises the use of techniques to select only the most relevant data for a specific analysis task;
- Data transformation - This is a process of transforming data by implementing aggregation or summary operations, thus transforming the data into an appropriate form;

- Data mining - This is done by extracting patters using different models. This part of the overall process is named data mining and in some cases can be considered as a sub part of KDD. Some sources consider data mining as the larger framework containing all of the processes and methods;
- Pattern evaluation - This comprises evaluation of patterns to determine if these patterns indicate knowledge;
- Knowledge presentation - This consists of visualizing and representing patterns or knowledge to relevant users.

4.4.2 Data Mining Process Model

CRISP-DM is a data mining process methodology first conceived in 1996 [4]. Angée [28] indicated that in 2014 CRISP-DM was still the most used methodology, but with decreasing interest and use. The CRISP-DM model has been refined and adapted into the Analytics Solutions Unified Method for Data Mining (ASUM-DM) by adding steps related to deployment and operations. In this dissertation, the process model is used to instantiate the ML models and the framework discussed in section 5.1 is used to add operational and deployment steps. Thus, the focus is placed on the CRISP-DM process model.

The following provides an overview of the process.

4.4.2.1 CRISP-DM

Figure 4.5 shows six different phases contained in the CRISP-DM model. These six phases have a suggested sequence, but depending on the outcomes of each phase, the sequence of execution can jump back to evaluate a previous stage. The large outer circle indicates that the process is a repetitive cycle that can result in more focused data mining tasks to improve existing models or to produce business knowledge that may lead to new dita mining tasks.

The following is a description of the different phases contained in the CRISP-DM process model [4]. These phases are describes as tasks and expected outputs.

- Business understanding: This phase consists of four different sub processes or tasks.

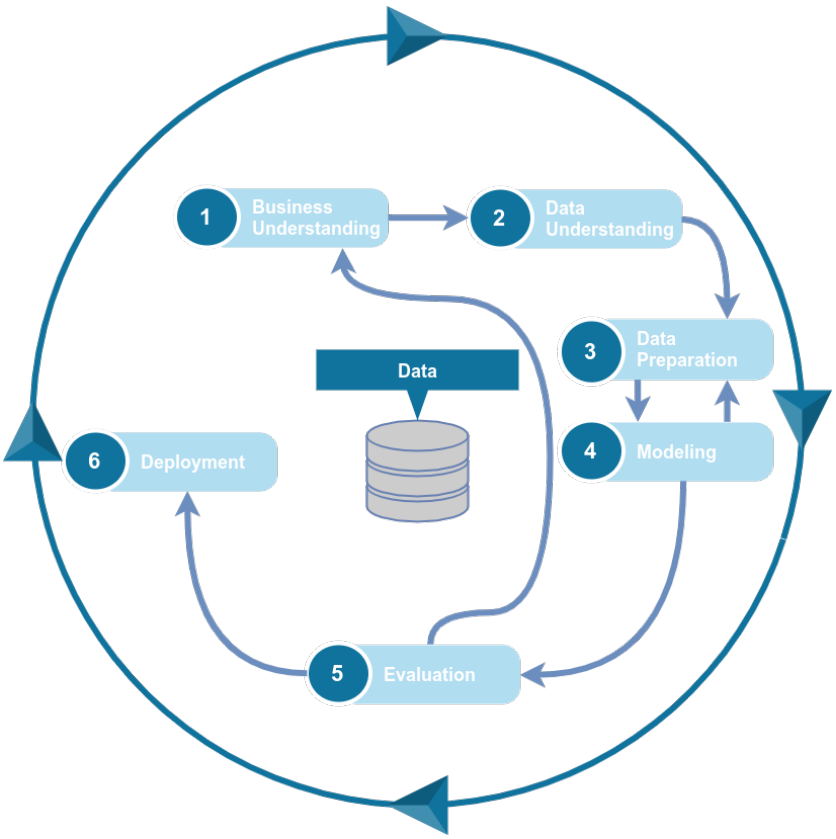

Figure 4.5: CRISP-DM model [4]

- Business objectives - A crucial initial step in the process is to understand the objectives and problems from a business perspective. This is important to align the business users' expectations with the data mining objectives. The outputs contain background on the relevant business processes, business goals and the criteria for evaluating the success of the data mining project;
- Situation assessment - This task consists of evaluating details of the required objectives. The outputs include determining available resources, detailed requirements, constraints and assumptions. This also includes determining risks and benefits involved with this data mining project, as well as defining the relevant terminology;

  - Data mining objectives - This requires converting the business objectives into data mining goals by expressing the objectives in technical terms and outcomes. The outputs of this task are the technical data mining goals and evaluation criteria that will be used to determine the success of the project;
  - Project plan - This task produces steps that will lead to the implementation of the data mining goals, including the required resources and duration, inputs, outputs and dependencies. This includes the assessment of different tools and techniques that can be used to achieve the data mining goals.

- Data understanding:
  - Collect data - This task is identifying relevant data sources and possibly loading the data into tools. The output is a report containing the data sources and all the required information to access and describe the data contained in the data sources;
  - Describe data - This task generates a detailed report on the format, size, quantity and other relevant properties of the data;
  - Explore data - This is done by evaluating the data to indicate possible key attributes and insights that can be gained by simplistic visualizations and aggregations or statistical analysis. This could possibly already satisfy the data mining goals;
  - Verify data - This task should answer questions regarding the completeness of the data, the number of missing values and the number of errors contained in the data.

- Data preparation: The main outputs of this phase is to produce the data sets and data set descriptions that will be used in the rest of the data mining process.
  - Select data - This includes the selection of relevant data. This can be based on business knowledge or evaluating the volume and data types. Different feature selection techniques can also be used;
  - Clean data - This requires removing or generating data points for missing or inconsistent data determined in the tasks above;
  - Construct data - This task consists of generating new features and is also known as feature generation [29];

  - Integrate data - This is the process of combining different data features into a new feature;
  - Format data - This is the task of changing the data to a format required by the tools or models used in the next tasks.

- Modeling:
  - Select technique - This consists of determining the appropriate technique or techniques that can be used to achieve the specified data mining goals. The tools or models available will be determined by different constraints and requirements. The constraints can include data format, quality or distribution. Requirements can include scalability, computational performance and model accuracy;
  - Generate test design - This includes the plans that describes the training, testing and evaluation techniques;
  - Build model - This consists of generating models (from data) by running the tools or programs that train the applicable models;
  - Asses model - This is evaluation of the model using the testing design plan. This step evaluates if the model meets the defined data mining success criteria.

- Evaluation:
  - Evaluate results - In this evaluation process the output of the models is compared to the defined business success criteria. This evaluates models as well as the findings produced by the models and data mining processes;
  - Review process - This is a task that reflects on the data mining steps. This includes generating a report on failures and gained insights;
  - Next steps - This is an evaluation process that determines if the project can be advanced to the development phase, or if additional iterations are required.

- Deployment:
  - Plan - This includes planning the deployment with regards to inputs and outputs, scalability, information propagation, etc.;

  - Monitoring and Maintenance - This is an important part of continually evaluating the success of the project. This includes evaluating changes in the data and goals;
  - Report - This task consists of generating a report that contains all of the relevant information generated in the data mining process;
  - Review project - This is a review that focuses on evaluating the business goals and success criteria.

The CRISP-DM process can be used as a guide to implement a data mining project. The steps described should be evaluated for relevance for each specific project and applied accordingly.

4.4.3 Machine Learning

Artificial Intelligence (AI) is a branch of study contained in Computer Science focused on creating machines that can act or react intelligently[30][31]. Machine Learning (ML) is an important part of AI.

ML can be defined as an automated method of detecting patterns or anomalies in data. The general approach of ML is to train a machine by providing training data to a learning algorithm that produces a meaningful output in the form of a trained model or the like [29].

There are three main aspects to machine leaning [29], as follows:

- Input: This can contain some or all of the following -
  - The data set containing the features that describe an observation of an underlying statistical process;
  - The labeled data set that contains a set of labels that describes an output that needs to be predicted applicable for supervised techniques, or for evaluating unsupervised models;
  - The training data containing a subset of the total data set to which a model will be fitted;
  - The test data containing a subset of the total data set used for evaluation of a trained model;
- Output: The output of ML is a predictor or classifier that describes a function or model used to predict or label new data points;

- Measure of error/success: This is an important part of ML and is used to evaluate a model. The data is usually divided into two separate subsets, of which the first is used as training data and the second as test data. The test data is used to calculate an error score of model's ability to predict or classify, based on previously unseen data, and to thereby evaluate the success of training.

ML is divided into three main categories, namely supervised learning, unsupervised learning, and reinforcement learning [32]. These are discussed below.

4.4.3.1 Supervised Learning

Supervised learning is the process of training a machine by using labeled data, which means that the training and test set include the target feature [32]. Thus, the model is trained with examples containing the expected output. After a model has been trained using training data, the model can then predict target labels (using similar features) from samples with labels that are unknown as these have not been previously encountered. The same can be done with a test set, namely to determine the difference between an expected output and the predicted output [29].

There are many different models that fall under supervised machine learning, which can further be divided into into two classes: (i) classification describes models that can be used to predict a discrete set of labels where (ii) regression describes models that can be used to predict a continuous set of outputs.

4.4.3.2 Unsupervised Learning

Unsupervised learning is typically used when a target feature is not included in the data set [32]. The general goal of unsupervised learning is exploration of data by generating a compressed version or summary of the data [29]. Clustering groups input data into groups of similar attributes, where the clusters are unknown beforehand, as opposed to classification where classes are known before training commences. These clusters can add information to the data that can lead to new, previously unknown insights - for example, if similarity had not been known beforehand, such groupings of data into similar classes may add meaning by way of association.

4.4.3.3 Reinforcement Learning

This ML technique, unlike supervised learning, does not include examples with the target feature. The techniques however do include a method of evaluating the best action or prediction by maximizing a reward value. This is achieved by using trial and error to interact with the environment and using feedback to optimize the model[32]. This is considered as a closed-loop method due to the fact that a decision made by the model influences the later inputs to the model [33].

4.4.4 Time series forecasting

Time series forecasting consists of predicting a future value of a time series based on past values in that series. A time series is a series of values that is obtained over a specified time interval or at regular time stamps. In general, a time series can contain four different components, namely: (i) trend, (ii) seasonal, (iii) cyclic and (iv) irregular or residual components [34]. Trend relates to a general increase or decrease in a time series. Seasonality relates to repeatable patterns that occur in the time series over a specific time frame, usually less than a year. The cyclic component relates to patterns that do not indicate a fixed period and usually spans periods of longer than a year. The irregular component relates to unpredictable elements in a time series [34]. Many different time series forecasting methods exist and extensive research has been conducted on these. De Gooijer *et al.* provides and extensive documented history on the developments of time series forecasting [35].

The following section provides an overview on 2 different time series forecasting methods of interest in this study. The first methods is a stochastic model called an ARIMA model. The second method is a neural network based method called an LSTM model.

4.4.4.1 ARIMA

An ARIMA model is a combination of two other models, namely the autoregressive (AR) model and the moving average (MA) model. The AR model predicts the next value as a linear combination of p past values, a random error and a constant. This can be described by the following equation AR(p) [35]:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t \tag{4.1}$$

where y_t represents the actual values at t, ϵ_t represents the random error at t, c represents a constant, ϕ_i $(i = 1, 2, ..., p)$ represents the model parameters and p the model order.

The MA model predicts the future value as a linear combination of past errors. The following equation describes a moving average model MA(q) [35]:

$$y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + ... + \theta_q\epsilon_{t-q} \tag{4.2}$$

where μ represents the mean of the series, θ_j $(j = 1, 2, ..., q)$ represents the model parameters and q the model order.

When the AR and MA models are combined with differencing an ARIMA model is obtained. A specific ARIMA model can be expressed with the following notation:

$$ARIMA(p, d, q) \tag{4.3}$$

with p the AR order, d the differencing order and q the MA order.

An adaption of the ARIMA model is a Seasonal ARIMA (SARIMA) model. This model removes non-stationarity from the seasonal time series using seasonal differencing of a specific order [34]. A specific SARIMA model can be described with the following notation :

$$SARIMA(p, d, q)x(P, D, Q)^s \tag{4.4}$$

with p,d and q indicating the orders for the non seasonal components and P, D and Q indicating the orders for the seasonal components. The s indicates the seasonal repetition interval.

4.4.4.2 LSTM

A LSTM (long short-term memory) network is a type of recurrent neural network (RNN), which in turn is a form of artificial neural network (ANN). An artificial neural network (ANN) is a mathematical structure containing artificial neurons and weights, structured in a manner representative of a human brain. That is, the network contains artificial neurons that may be linear or non-linear that are interconnected by weights that functionally resemble axons and dendrites. An ANN may adapt its interconnection paths by means of mathematical optimization, that is, the network effectively learns by means of minimizing the error between calculated and pre-recorded outputs [34]. The model thus contains a network of interconnected “neurons”that can produce a complex non-linear transfer function between inputs and outputs in a multi-dimensional space. An RNN is a neural network that incorporates feedback

from outputs of neurons to their inputs, thus producing an internal state and providing a type of temporal memory. This makes RNN networks ideal for sequential data [36]. This “memory”is achieved by repeating a number of individual structures or modules in a larger chain-like structure [37].

A LSTM is a type of RNN that allows for incorporating long term and short term dependencies. The internal structure of each LSTM cell contains additional gates that essentially determines how much of the input should be remembered, when a value should be forgotten and how much of a value should be included in the output [38].

4.4.5 Anomaly Detection

An anomaly can be seen as data behavior that differs significantly from a well defined normal pattern of behavior [39][40][41]. Anomalies often indicate critical actionable information and it is thus important to be able to detect anomalies. Anomalies do not always indicate a negative event but can also indicate a positive event[40].

There are three different anomaly types:

- Point anomalies can be defined as single points of data that differ significantly from the containing data set [39]. An example of a point anomaly is shown in Figure 4.6;
- Contextual anomalies can be identified as anomalies due to the context of the data. The data generally consists of two attributes, namely (i) the behavioral attribute that indicates the actual value or anomaly and (ii) the contextual attribute that provides a context to a data point [39][41]. An example of a contextual attribute can be spatial information or a time stamp for a time series. Figure 4.7 indicates an example of a time series with a clear deviation in its pattern (or its seasonality), this indicates that the point shown can be considered as anomalous due to the context provided by the seasonality (repetitive nature) of the series;
- Collective anomalies can be defined as anomalies that occur when a group of data points indicate an anomaly while a single instance would not indicate an anomaly. Figure 4.8 indicates possible collective anomalies. An example of a collective anomaly is when a large group

of devices generates slightly abnormal but acceptable data at the same time. The data of a single device can be considered as normal, while the data generated by a group of devices at the same time, can be considered as a collective anomaly.

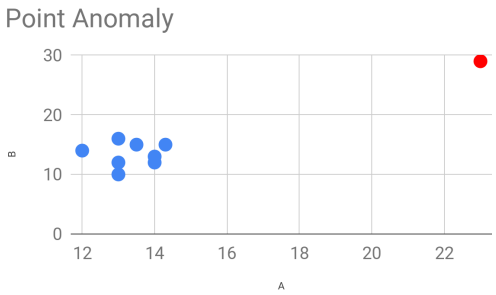

Figure 4.6: Point Anomaly Example (Anomaly shown in red)

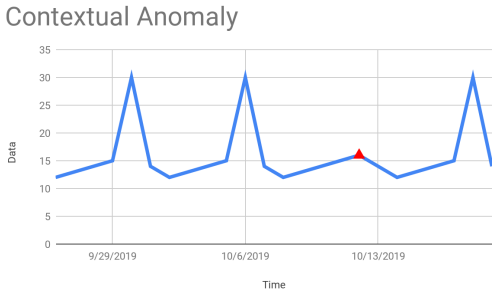

Figure 4.7: Contextual Anomaly Example (Anomaly shown in red)

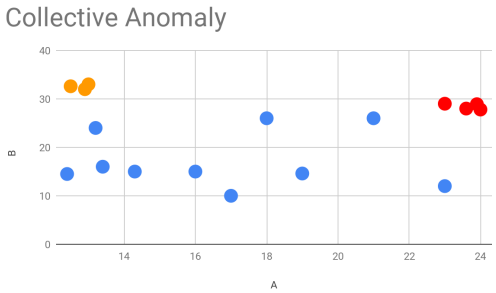

Figure 4.8: Collective Anomaly Example (Anomaly shown in red)

A wide range of different techniques and approaches to anomaly detection exists [42][40][43][44], of which to are of interest in this study, as described below.

One time series anomaly detection technique consists of predicting a time series and evaluating the difference of the predicted value to the actual value. Improving the prediction accuracy could in turn improve the anomaly detection accuracy, thus focus is placed on the time series prediction techniques.

Clustering can also be used as an anomaly detection method. This groups data into clusters with smaller cluster possibly indicating anomalous data.

4.4.6 Model Performance Evaluation

Evaluation of models is a crucial aspect of data mining, where the evaluation process determines the success of a model. This is used to compare models to determine a most suitable model and to determine if its accuracy is acceptable. There are two main classes of evaluation methods, namely (i) classification problems with discrete predictions, and (ii) regression problems with continuous predictions. Most evaluation techniques require labeled data, or the ground truth, in order to compare a predicted value to an actual value.

4.4.6.1 Evaluation of Classification Performance

Classification prediction evaluation models generally calculate the error or accuracy of a model using a combination of the number of true positives (TP) (predicted value is 1 and actual value is 1), true negatives (TN) (predicted value is 0 and actual value is 0), false positives (FP) (predicted value is 1 and actual value is 0) and false negatives (FN) (predicted value is 0 and actual value is 1). The combination of these values is grouped and presents a confusion matrix [45].

Commonly used performance indicators include:

- Accuracy - this is the number of correct predictions over the total number of predictions. Accuracy can be misleading when the data-set is class-imbalanced, meaning that one of the classes, positive (1) or negative (0), contains most of the observations;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

- Precision - this measure indicates the proportion of true positive to all positive predicted observations;

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

- Recall, sensitivity or true positive rate (TPR) - this measure indicates the proportion of true positives to the total number of positive observations;

$$Recall = \frac{TP}{TP + FN} \tag{4.7}$$

- Specificity or true negative rate (TNR) - this measure indicates the proportion of true negatives to the total number of negative observations;

$$Specificity = \frac{TN}{TN + FP} \tag{4.8}$$

- False Positive Rate (FPR) - this measure indicates the proportion of false positives to the total number of negative observations;

$$FPR = \frac{FP}{FP + TN} \tag{4.9}$$

- Receiver Operating Characteristics plot (ROC) - this is a graphical plot of the TPR against the FPR at different model threshold values. This provides a visual aid indicating the trade-off between the TPR and the FPR [27].The area under the curve, referred to as the AUC, indicates a scale-independent measure of the performance of the model. A value of 0.5 is considered as the baseline model and 1 is considered to be a perfect model;
- Precision-Recall curve (PRC) - this is a visual representation of the precision against the recall. The PRC can provide a better representation of the fraction of true positive prediction contained in all of the positive predictions. This is preferred to the ROC curve when dealing with imbalanced data sets as the ROC can be misleading [46]. The AUC of the PRC can also be used to evaluate the model-wide performance. It is important to note that the baseline for the PRC is a horizontal line with height equal to the proportion of positive examples in the data ($y = P/N$, P = Positive examples, N = Training data size) [46];

- F_β-score - The F_β-score is a combination of the precision and recall and defined as:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} (0 \leq \beta \leq \infty) \tag{4.10}$$

with P indicating the precision and R indicating the recall.

The β parameter controls the weight between precision and recall. A β larger than one ensures that recall has a stronger influence on the score [47]. This is valuable when one of these metrics is more important to the success of the model.

4.4.6.2 Regression Evaluation

Evaluating a regression model usually consists of calculating the error between the predicted values and the actual values. Multiple different evaluation techniques or error measures exist. Hyndman *et al.* classifies these measures into four classes, namely: (i) scale-dependent measures, (ii) percentage based error measures, (iii) relative error based measures and (iv) relative measures [48].

The following list provides a few common regression performance measures.

Variables:

- N - Number of observations;
- y_i - i-th observation;
- $\hat{y}_i$ - i-th prediction.

Measures:

- Mean squared error (MSE) - this error measure is one of the least complex and calculates the average of the squared error. This is classified as a scale-dependant error measure. It is however important to note that the error dimension is the square of the original dimension. The accuracy of the predictions increases as this value decreases [49];

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4.11}$$

- Root mean squared error (RMSE) - this metric is the square root of the MSE. This is classified as a scale-dependant error measure. The square root is used to convert the dimension of the error value to the same dimension of the original values. The accuracy of the predictions increases as the value decreases [50];

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} = \sqrt{MSE} \tag{4.12}$$

- Mean absolute error (MAE) - this is a average of the absolute difference between the predicted and actual values. This is a scale-dependant measure. This measure is also a linear score compared to the RMSE, which penalizes larger values exponentially [51];

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.13}$$

- R-Squared (R^2) - this metric, also called the coefficient of determination, is a dimensionless measure and thus scale-independent. The maximum value of the R^2 metric is 1 and the minimum is $-\infty$. The accuracy of the model increases as the R^2 value increases to 1. The R^2 model is in comparison with a naive model that predicts the value as the average of all values;

$$R^2 = 1 - \frac{MSE(model)}{MSE(baseline)} \tag{4.14}$$

with

$$MSE(baseline) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2 \tag{4.15}$$

with $\bar{y}_i$ the mean of the actual values.

- Adjusted R-Squared (R^2) - this metric adjusts the R^2 metric to consider the number of terms in the model. This means the adjusted R^2 metric penalises metrics that do not add information to the prediction:

$$R^2_{adj} = 1 - [\frac{(1 - R^2)(n - 1)}{n - k - 1}], \tag{4.16}$$

with n the number of observations and k the number of predictors.

- Mean absolute percentage error (MAPE) - this metric is percentage based and can be considered as scale-independent [49]. As the percentage decreases the accuracy of the model increases;

$$MAPE = \frac{100\%}{N} \sum_{i=1}^{N} |\frac{y_i - \hat{y}_i}{\hat{y}_i}| \tag{4.17}$$

- Symmetric mean absolute percentage error (SMAPE) - this metric is an adaption of the MAPE metric. The SMAPE metric penalizes positive and negative errors equally, unlike the MAPE metric [48];

$$SMAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \tag{4.18}$$

- Mean absolute scaled error (MASE) - this metric is scale-independent and values greater than 1 indicate a model predicting worse than a naïve one-step forecast method. Values less than 1 show an improvement on the naïve method;

$$MASE = \frac{\frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|}{\frac{1}{N-1} \sum_{i=2}^{N} |y_i - y_{i-1}|} \tag{4.19}$$

These measures have different advantages and disadvantages, depending on the application [51][35][49]. The scores are included in the discussion above for the sake of completeness.

4.5 Systems Engineering

This section provides an overview of Systems Engineering (SE) as it forms the basis for determining the BI operational framework. The importance of a full life-cycle approach to SE is defended and a description of the different phases of a system is provided to this effect. Finally the SE process is summarized as a set of steps or processes that can be executed in a iterative manner. The analysis of other BI frameworks provides input requirements as the basis for the framework presented in this research. In short, SE provides a detailed process that incorporates a full life-cycle approach to implementing a system, this provides valuable input that can be used to define a BI framework as is described below.

4.5.1 System Definition

Blanchard *et al.*[5] mentions that the term "system"is used in different contexts and situations. In order to ensure that the definition of SE is fully understood, the definition of a system is required. Blanchard *et al.*[5] summarizes the definition of a system as follows:

> "A system constitutes a set of interrelated components working together with the common objective of fulfilling some designated need."

They continue by describing a system based on the characteristics and different categories of systems.

The characteristics of a system are as follows [5]:

- Resources - A system requires different resources to be combined and utilized in an effective manner. These resources include materials, software, hardware, equipment, human beings, documents, data etc.;
- Hierarchy - A system forms part of an overall hierarchy with external factors and higher-level parts influencing the performance of the system;
- Subsystems - A system, depending on complexity, can usually be broken down into different subsystems and components. This allows for evaluating each subsystem individually and the interactions between these subsystems and components. It is important to note that the system needs to be considered as a whole before each subsystem is determined and evaluated;
- Purpose - Each system needs to have a purpose. This means that the system is functional, resolves a specified need and achieves a set of determined objectives in a cost-effective manner.

There are different categories under which a system can be grouped. These categories are defined to show the wide range of different systems, including:

- Natural systems compared to man-made systems;
- Physical systems compared to conceptual systems;
- Static systems compared to dynamic systems;
- Closed-loop systems compared to open-loop systems.

Blanchard *et al.*[5] further mentions that there are 4 high-level components connected to a system. These are its inputs, outputs, constraints and mechanisms as shown in Figure 4.9.

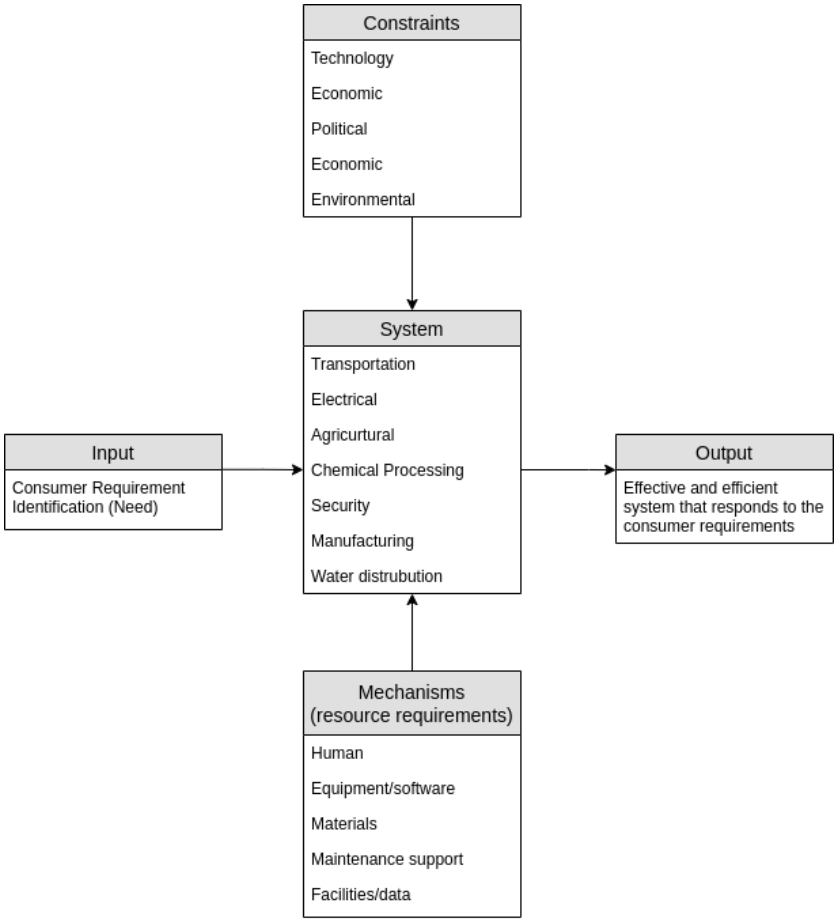

Figure 4.9: High level system [5]

System mechanisms can be categorized to fall under either (i) system elements or (ii) enabling system elements. System elements contain operational resources, including operational staff, operational equipment, configuration information, operational software etc. Enabling system elements contain the resources required to enable design and implementation of a system as well as maintenance and disposal of a system.

A System of Systems (SoS) is a term describing different layers and component systems that collectively achieve the overall goal of the entire SoS. These systems can have individual operational goals or objectives but contribute to a higher level mission. It is important to define the different systems, including their hierarchical relationships, the boundaries of the systems, and interfaces between these systems. This provides a better understanding of the entire system or SoS.

4.5.2 System Life-cycle Phases

A system full life-cycle consists of all relevant life-cycle phases of a particular system, following a specified "cradle-to-grave"process. This allows identification of all required resources, and allows planning for the full life-cycle to limit unexpected resource expenditure. The different life-cycle phases are described below[52]:

- Development - Includes all of the activities required to produce a final system or product that meets the requirements elicited from the initial needs. This phase is discussed in further detail in the section below;
- Manufacturing, production and/or construction - This includes activities and resources required to produce or fabricate the product or to construct the system or subsystems;
- Deployment - This includes activities and resources that support the implementation of the system. These include delivery, transportation, storing, installation etc;
- Operation - This is the activities required to achieve the main objectives of the system;
- Maintenance and support - This includes maintenance operations, logistics, operational support required to maintain the system operations;
- Disposal - This includes the activities to ensure that the end of life operations for different components in the system is defined and implemented according to specifications;

4.5.3 System Engineering Process

SE is defined in different ways, but essentially comes down to the same definition [5][53][54]. SE can simplistically be defined as an interdisciplinary approach aimed at realizing a system. SE is thus a high-level method that implements a top-down, iterative process considering all different phases in an effort to transform an operational need into a cost-effective solution (system).

Five important factors of system engineering are emphasized by Blanchard *et al.* [5]. These are that:

- SE requires a top-down approach. This ensures the system can be evaluated as a whole and a better understanding can be gained of the interaction between sub components and the hierarchy of the system under evaluation.
- A full life-cycle view of a system is required, which ensures focus is placed on all of the different life cycle phases of that system individually and as a whole.
- The system requirement identification step requires sufficient effort to ensure that all of the specific design goals and criteria is defined fully, reducing later complications and unnecessary resource expenditure. This is done in conjunction with requirements management throughout acquisition phase of a system.
- Interface management is essential to manage planning, monitoring and implementation of all levels of integration activities.
- Finally, technical performance management is required for verification purposes (as part of an overall validation and verification process), including resource allocation management.

The phases and activities included in the SE life cycle process can be summarized in Figure 4.10.

The following section describes the most relevant aspects of the SE process. The phases follow the order provided, and some of these phases can be done in parallel. In some cases it is necessary to iterate over phases, thereby increasing the quality of the output of each of the phases over time. The overall process is followed to implement the initial system, but when new requirement are introduced, repetition of the phases is required.

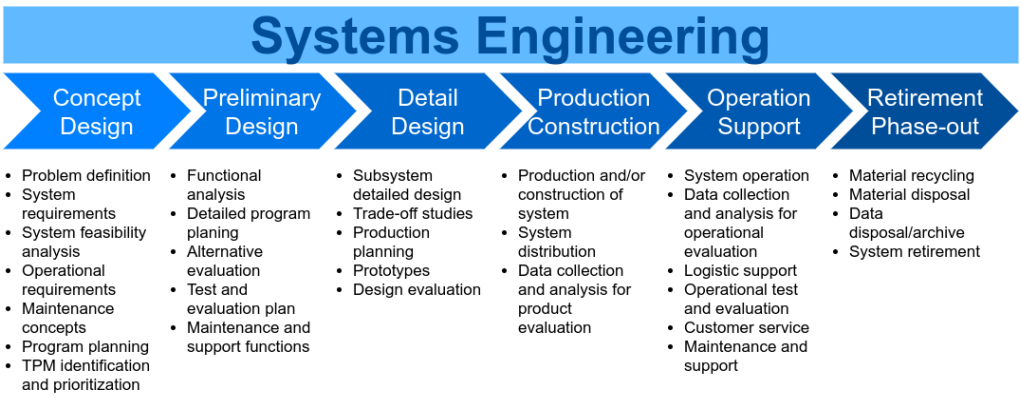

Figure 4.10: Systems Engineering Process [5]

4.5.3.1 Definition of Need

This is the first step in the SE process and provides a definition of the need that the system has to address. It is essential to provide a good definition of the need as this allows better definitions of consequently derived requirements. It is also important to define the risks associated with the need at hand. This process requires good communication with the client to define the actual need or problem of the client and to reduce the risk of unnecessary resource expenditure or unmet expectations.

This section also includes the definition of high-level system requirements. This is a process of converting the problem or need into specific requirements by defining the functions needed to address the challenge. Primary and secondary functions are usually defined and include constrains like when, where or by whom a function is required.

4.5.3.2 Concept Design (Requirements Analysis)

After the high-level system requirements have been defined and a feasible study has been done on a concept design, a comprehensive description of the operational requirements of the system is required. These requirements convert the need of the client into operational requirements that describe the deployment and distribution, different operational scenarios, performance requirements, utilization requirements, effectiveness requirements, interface requirements, environment requirements and other requirements necessary to evaluate the success of the system over a full life-cycle.

It is important to not only focus on the operational performance of the system but also on the aspects related to maintaining the required performance across the full life-cycle of the system. The maintenance concepts identified here should be considered throughout the rest of the design and will affect the choice of solution. This should also be used as a guide to create the detailed maintenance support and logistics plan.

In the definition of the BI framework, it is important to note that the requirements management process is important as high level business requirements and objectives must be met by the BI system. Also, maintenance is critical as no system will function without maintenance and ongoing system optimization. These two principles are used in the BI framework.

4.5.3.3 Preliminary and Detail Design, and Integration

This is the technical design phase and should provide, as output, the final detailed design. The concept design and requirements from prior phases are used as input to this phase.

A functional analysis is an essential and important task in the conceptual or preliminary design and deserves special mention. This includes definition of all functions required to implement a successful system as well as a breakdown of the main functions or requirements into all of the lower level functions required to achieve the system goals and objectives. There are two main categories of functions, operational functions and maintenance functions as discussed in section 4.5.2.

An initial form of a functional analysis can be implemented to support the definition of the operational requirements and maintenance concepts. A later, more detailed iteration can then be implemented to define the full system functional analysis.

Functional flow diagrams are used to present the different functions and interfaces between functions. The top-level functions are defined first and then partitioned into sub levels until the required level of definition has been met.

Integration is done by initially completing a comprehensive system breakdown, supported by a system architecture with all functional blocks and interfaces. This is followed by integration planning as all interfaces are defined by two interacting resources (functional blocks) that must be initially separated and later integrated. The integration effort is then managed throughout the process from the bottom upwards as lower level functions are being implemented.

The use of functional analysis principles is important to the definition of a BI framework process as it is used as a tool to analyse and visualise this process. Furthermore, integration of models and methods is also important and the importance of management of the integration effort is underlined.

4.5.3.4 Test and Evaluation

Full evaluation of the system can realistically only happen when the system has been finally integrated just before the operational phase. However, ongoing testing and evaluation of subsystem components can provide valuable feedback that can save costs if changes are required. The goal of testing and evaluation is to gain confidence that the system will achieve the expected goals or objectives while adhering to specified requirements.

Testing occurs throughout the life-cycle of the of the system. These tests can be categorized into five different types:

- Analytical test that can be achieved early in the design phase (simulations, analytical models, computerized techniques) and usually occurs in the conceptual design phase;
- Type 1 tests evaluates different models, mock-ups and early prototypes and usually occurs in the preliminary design phase;
- Type 2 tests evaluates prototype and production models and usually occurs in the detailed design phase;
- Type 3 tests the production models that have been implemented and occurs after the production/construction phase;
- Type 4 tests are continuous evaluation of the operational system.

The test phase discussed here is important to the BI framework process as the process incorporates significant test effort.

4.5.3.5 Production, Operation, and Support

The structure and flow of this phase is heavily dependent on the nature of the realized system and the process is discussed here for the sake of comprehensiveness. In information technology terms, “production systems”are used in production environments and describe the BI framework when deployed and in operation.

This phase also ensures that the implemented system continues to perform at an expected level. This requires continuous testing and evaluation, training and support, maintenance of the product, evaluating performance and improving the system where necessary. Collecting, storing and providing access to operational data for analysis is a key part in this phase.

As the BI framework include provision of a solution in the form of models, it is imperative to understand that the “production system”must be maintained and optimized when in operation.

4.5.3.6 Retirement and Disposal

The retirement of a system not only affects the physical components and materials but also includes the documentation, data and intellectual property of the system. A retirement or phase-out plan is required that details the procedures involved with the different components of the system when the system reaches this phase of the life-cycle.

In the case of the BI production system, the system will most probably be ported to a new platform at end of life (if necessary) or upgraded. Functions will most likely remain relevant, but the implementation methods may differ, implying that the system will have to be re-engineered when major changes are required. This will be done by following the phases from the Concept Design phase onward, and will be using historical documents, data and results in the form of a Body of Knowledge.

4.6 Conclusion

This section concludes the literature review conducted for this research study. The literature was studied according to the Literature Focus Areas to inform the research challenges discussed in Chapter 3. These Focus Areas also contributed to validate the research challenges and to provide valuable insights and understanding used to inform the research solutions. This is summarized in Table 4.1 as discussed below.

- Area 1: The architecture of a general IoT system was provided and discussed as well as the different layers and data flow in an IoT system. This provided insights into how BI can be incorporated into an IoT system, and also provided valuable understanding of IoT - this was included in a general BI ontology. Data is an integral part of an IoT system and utilizing this data to gain insights is a crucial part in improving system performance, this validates the need for BI in IoT.
- Area 2: Research on BI provided a good understanding and definition of BI and this was also incorporated into the BI ontology. The literature provided insight into existing frameworks and challenges, validating the need for an integrated implementation and operational BI framework focused on an IoT environment. The BI literature study thus provided valuable input that supports implementation of an integrated BI solution.
- Area 3: Data mining and machine learning were researched and information on the CRISP-DM process methodology was provided, different machine learning techniques and process were explored and also discussed. This provided valuable information incorporated into the BI framework provided in Chapter 5 and the different models developed in Section 5.3. The research done on machine learning models, anomalies and evaluation techniques allowed exploring, evaluating and implementing different anomaly detection techniques in the overall BI framework in this research.
- Area 4: The literature study on cellular communication theory provided a fundamental understanding of the characteristics of a cellular IoT system. This aided in identification of the characteristics of the system under evaluation, namely a cellular IoT network. This study thus provided important insights into the different measurable status indicators provided by a cellular IoT system.

- Area 5: Systems engineering (and thinking) forms the basis of the BI framework and supports the importance of a full life-cycle approach to implementing a system. The definition of a system and importance of different aspects of the SE process provided validation of the value of identifying system characteristics. Furthermore, the use of requirements management, functional analyses tools, validation and verification, system optimization and maintenance all form part of the SE principles - these principles all aided in the development and implementation of an integrated BI solution.

Table 4.1: Literature focus areas

		Research Challenges			
		Lack of implementation framework for BI in cellular IoT	System characteristics unknown	Lack of intelligence ontology	Lack of Integrated Application
Focus Areas	IOT	↑↓		↓	
	BI	↑↓		↓	↓
	Machine Learning and Data Analytics		↓		↓
	Cellular Communication Systems		↓		
	Systems Engineering	↓	↑↓	↓	↓
		Definition of BI framework for cellular IoT	Definition of system characteristics	Definition of intelligence framework ontology	Integrated system perspective
		Research Solutions			

Chapter 5

Synthesis

This chapter provides the definition of a BI framework, and can be divided into two main sections, namely (i) a section defining a general BI framework model that can be used to implement a BI platform (solution), and (ii) a section on implementation of a BI solution for an existing real word IoT system using the BI framework model.

The BI framework provides a generalised architecture of a BI system, which aims at providing a better understanding of the different sections and components of a BI system. Also, a process flow model is used to provide a logical flow of tasks to be followed when implementing a BI solution. The Development and Operations phases are expanded as the core components of the framework. The Development phase of the framework realizes a BI platform that allows business users to effectively extract insights from available data. The Operations phase of the framework includes a process flow model that can be used by different users to extract insights using this BI platform.

5.1 BI Framework

5.1.1 BI Architecture

BI is a system that consists of a range of different processes, methods, data, tools and users. This system is always part of a larger system and can thus be considered as one of a SoS.

The BI system architecture can be divided into three main sections, including Data Management, Data Analytics and Data Visualization as shown in Figure 5.1. Each of these can be subdivided into different tools, data elements, methods, processes and users. Not all BI implementations will require all of these sub-components and implementing a BI solution will require evaluating the system to determine the required components.

The following provides an overview of each component and describes the sub components and processes included in each. It is important to note that different BI vendors provide solutions with varying complexities. Some vendors provide a full BI solution and others provide specific components or tools that can be integrated into a BI system. The selection of these tools and components will determine the processes and actions required to implement and maintain the system.

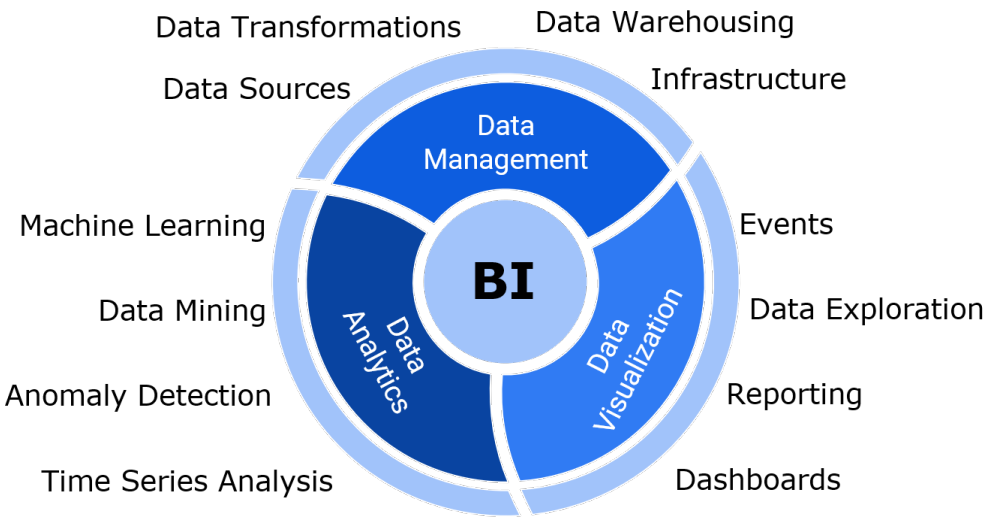

Figure 5.1: BI Conceptual Framework

5.1.1.1 Data Management

This section contains all of the data storage components and operations that convert the data into a useful form.

- Data collection - This is the process of identifying relevant data sources and incorporating these sources as the data sources of the system. This can include loading data into a data warehouse or creating a connection that allows tools to access the data directly from the source or through an ETL process;

- Data transformation - This consists of converting the data into a more appropriate form. Different tools might require the data to be formatted into their respective formats. Most BI tools require data to be in a relational form and thus the transformation of this data might be required. This could also be achieved, in some cases, by drivers that convert the data into such a relational format. The transformation can occur before the data is saved in a database and made available to conserve computational processing resources, or the data can be converted as the data is requested in order to conserve storage resources;
- Data maintenance - This is the process of evaluating the data for inconsistent or missing values. This includes reviewing the data to evaluate the validity of storing the data, thus determining if the data is required, and if the data adds value;
- Infrastructure Management - Scaling and infrastructure management is a crucial part of this unit. Different online vendors provide a variety of online tools and infrastructure to support these operations.
- Data sources - This includes internal and external data sources. The data sources can be connected directly or a ETL process can be used to extract and save data from the source;
- Data transformation tools - this is software tools or techniques that transform the data into an appropriate from;
- Data warehouse - This is to separate the BI data from the operational data and infrastructure to ensure normal operation of the system is not affected. The data can also be transformed and loaded directly into the other units, without the data warehousing step, if the data and infrastructure allow for this.

5.1.1.2 Data Analytics

This operational cycle of the BI system contains methods used to explore data to detect trends and patterns, or to evaluate a specific problem. The output must then be provided as actionable insights using a data visualization component. This task employs a wide variety of methods, including machine learning and statistical methods. The main operations consist of data mining to detect or predict different trends an anomalies. This is process is described in more detail in Section 5.1.2.2.

This section mainly focuses on creating models that use data to detect or predict certain events or metrics linked to key performance indicators (KPIs). The output from here feeds into the visualization unit to provided useful access to relevant business users.

It is important to note that all involved business users should be included in the data analysis processes of this component in order to incorporate much as possible relevant business knowledge.

Different analytical tools can be used to execute the tasks required to provide AI models from data. Cloud computing vendors provide various tools, of which some are more intuitive and user friendly. User friendly tools provide a graphical interface to execute a variety of analytical tasks, but may be limited in terms of flexibility. Other less user-friendly tools provide more custom features and better scaling solutions, but require more experienced users as the interfaces are more complex.

5.1.1.3 Data Visualization

This component contains the interface that the business users use to access the BI system. This component consists of four main elements.

- Dashboards - Dashboards usually contain customized graphs and figures that indicate the current status of the system. This includes performance metrics and risk indicators;
- Event Alerting - This element consists of methods used to alert specific users when a certain event occurs. The cause of an event can be the output of a specific model or exceeding a basic threshold on a KPI, for example. This alert usually contains critical information;
- Data Exploration - This element provides a visual connection to the data sources. This usually consists of customizable graphs created by selecting different data sources and metrics, comparing different dimensions and aggregations of the data, and drilling down into the data to explore events, anomalies and trends;
- Reporting - This consists of creating custom shareable reports of historical data, aggregating the important metrics into a summary of the relevant data for a specific time.

5.1.2 BI Life-cycle

This section describes the life-cycle of a BI system. A SE approach is followed with specific focus on the Development and Operation phases in the sections to follow. A summary of the phases are as follows:

- Development phase - This phase consists of the processes and tools that enable designing and implementing the BI system. This section is described in section 5.1.2.1;
- Operational phase - This phase consists of the operational process that consists of generating insights from the available data. This section is also described in section 5.1.2.2;
- Maintenance and support phase - This phase comprises two main sections. The first is continuous evaluation of the system while in operation to evaluate if improvements or maintenance are required on the system. The second is training to enable users to correctly use the tools provided by the BI system;
- Retirement phase - This phase is less intense (but not less important) and will only occur when the entire system reaches end of life. It is important to determine and define the data storage and archive requirements if the system is to be retired. This will reduce unforeseen costs and data loss (including loss of a body of knowledge).

5.1.2.1 Development Phase

The Development phase of the framework includes the processes required to implement a BI platform. This platform is then used in the Operational phase to execute the main functions of the BI system. The development phase is based on the SE process, this is reduced to five steps shown in Figure 5.2.

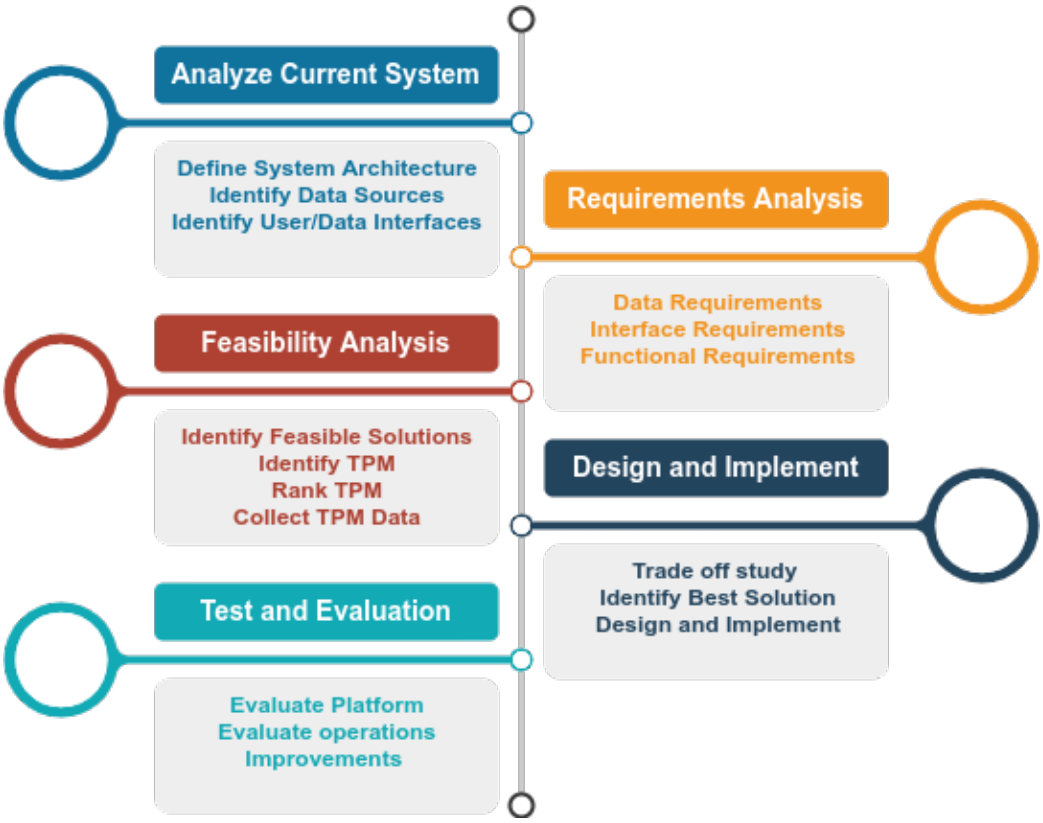

Figure 5.2: BI Development Process

5.1.2.1.1 Analyze Current System

As the BI solution is implemented to support and improve an existing IoT system, it is important to first analyze the current system. This is achieved by defining two main elements:

- The first is to define the current system architecture allowing for a better understanding of the entire system and the components of the system. The focus of this analysis should be to define the different layers in terms of user interfaces and user functions as well as the infrastructure and functionality of the system with regards to data management and data visualization.
- This second is to provide a detailed description of the available data sources that the system generates. This includes access methods, storage details, format, attributes and any other details that is required to define the available data.

5.1.2.1.2 Requirements Analysis

Requirements Analysis is focused on defining the requirements of the different system components. These include the data management requirements, data analytics requirements, data visualization requirements and also security and access requirements. This section should include short term and long term operational goals or objectives that can be used to define requirements, as well as the definition of the operational validation process. These goals must provide an indication of the complexity of the required analytics components. For example, do the goals require complex prediction and detection, or would business rules and appropriate data visualization suffice.

The following provides a basic description of requirements:

- The data requirements can include aspects such as supporting external sources, database requirements, big data requirements, infrastructure, transformation functionality and so on.
- Data analytics requirements relate mostly to the platform and tools used for analytics. This is determined by the different features required to achieve the operational goals.
- The data visualization requirements include defining the requirements for dashboard, alerting, reporting and drill down capabilities.
- The security requirements should define the different data access levels across the entire BI platform for the relevant users.

This step will provide an understanding and initial definition of the different components that are required.

5.1.2.1.3 Feasibility Analysis

In this section the technical performance measures (TPMs) are determined for the different aspects or components of the BI platform. These TPMs can be ranked according to importance. Solutions should be identified that meet the relevant requirements defined. These solutions can be explored to identify and collect the relevant data needed to indicate the TPM of a specific technology solution.

A BI solution can consists of integrating a range of different technology solutions. This can include developing some components in house, integrating cloud based services or purchasing a full solution from a BI consulting company. The BI tools available range in complexity, cost, features and usability. The current system evaluation could provide details about available resources and could also produce some constraints. This should be considered when evaluating feasible solutions.

5.1.2.1.4 Design and Implement

This task provides the definition of a detailed design, consisting of the solution architecture that incorporates the components and interfaces. This task also includes a detailed analysis of the different functions that will be included as part of the solution.

The feasible technology or design solution needs to be evaluated using technical performance measures and as appropriate method as described in Section 4.5.3.3. The top down approach of systems engineering should be incorporated evaluating the solutions from a full life-cycle perspective. This includes required support, maintenance aspects and the ability of the solutions to scale as the IoT system expands and grows.

5.1.2.1.5 Evaluate Platform

Testing and evaluating the platform consist of evaluating the different components to ensure that the requirements defined for each component are met. This is equivalent to Type 3 tests described in Section 4.5.3.4. This can occur as the system progresses through implementation and integration.

This process also includes testing the performance of the platform in the Operational phase as described in Section 5.1.2.2. This should be done by implementing one of the short term goals. This is equivalent to a type 4 test described in Section 4.5.3.4. This provides feedback on the different component performances and indicates where improvements must be made.

The last part of the Evaluation phase is the continued evaluation of system performance while in the operational phase (i.e evaluation of the “production system”). This is important to ensure that data and insights are reliable, and that the platform is continuously improved to meet expanding requirements.

5.1.2.2 Operational Phase

The goal of the Operational phase of a BI system is to iteratively extract insights from data and convert these insights into actionable intelligence. This section describes a process flow model (in the form of a functional flow) to achieve this goal. The process flow model is based on an Agile approach, which in turn is based on the CRISP-DM process with an additional process flow layer implemented on top of the CRISP-DM process.

The focus here will be on the process flow layer layer and not on the CRISP-DM process that was described in Section 4.4.

The data exploration task is the main function of the BI operational phase. The initial task discovers insights and determines what the next steps in the process flow should be, this is shown in Figure 5.3. This task is executed by multiple users with different levels of business knowledge to allow discovery of a range of different insights.

In general, this task can be divided into two main approaches, namely (visually vs. analytically):

- Where business knowledge is combined with data visualization to gain data understanding and to obtain insights - this is usually executed by a domain specialist and indicated by process reference 1 (P/R 1) in Figure 5.3.
- Exploring the data analytically with the goal to discover informative patterns and then to include business knowledge to obtain insights - this is usually executed by data scientists indicated by P/R 2.

After insights have been obtained from data, these insights should be evaluated to determine which of the following steps should be executed. The process is shown in Figure 5.3 and the options are:

- The insights should be included in the business knowledge as a form of intelligence (P/R 3), for example detecting and removing the cause of a single anomalous event does not necessarily require any additional model or business rule creation, but provides information that could improve the overall system performance;
- The insights typically lead to new KPIs that can be extracted or aggregated from data, to be included as an information source in the BI platform (P/R 4). The identification of KPIs in itself is useful as these can be used on informative dashboards or the like;

- If the insights relate to events that must be automatically predicted or detected, a CRISP-DM process can be initialized to create models that can perform such automated prediction and classification (P/R 5). The output of this task is a set of deployable classification and prediction models that provide predictions - these are then integrated onto the client's BI platform in the real-world.

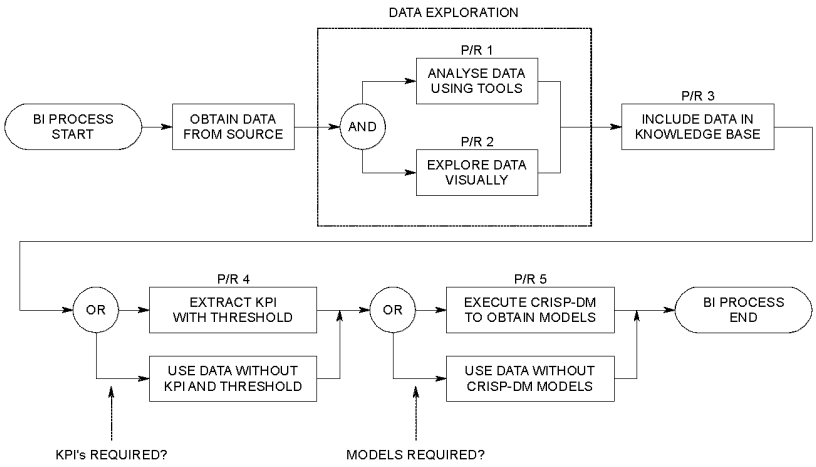

Figure 5.3: BI framework process flow model

In the section that follows, the development of a BI platform for an engineering company is discussed using the BI framework developed here (specifically following the BI framework process flow model). This engineering company focuses on IoT networks (cellular) and requires a BI platform onto which BI can be implemented. As a result, the process flow model described above could be used to provide a BI platform with which to achieve this purpose. The purpose of the case study below is thus to show the value of using the BI framework process flow, and to execute a number of experiments that correspond to the tasks shown in the process flow model.

Figure 5.4 gives an overview of the BI framework and the architecture in which the BI platform is included. The diagram indicates client edge devices that generate field data and communicate to the client's existing cloud platform. This data is accessed by the BI platform, aggregating and transforming the data into relevant formats to be accessed using the tools described below, with the goal to extract insight from the data. These tools, as discussed in the sections that follow, were all developed by the design researcher in order to provide a BI platform solution.

The BI platform includes multiple functional tools as described below:

- Visualization tools - This includes customizable dashboards that visually indicate real-time data related to the main KPIs of the system. The visualization tools also graphically represent different system metrics to provide drill-down capabilities and exploration of data in order to investigate different scenarios;
- Manual labeling tools - These enable the labeling of anomalies discovered by using the visualization tools described above. This also allows validating existing models, training new models, and creating and extracting labeled data sets.
- Reporting tools - These provide historical reports containing aggregated KPIs or events over a specified time period.
- Event alerting tools - These tools escalate events from the trained models that were generated using the BI framework process flow (the process followed to fit models to data in an iterative way as described above). The tools also perform non-critical notifications where required.
- Model-based tools - These are tools that provide real-time predictions and classification of events as obtained from running the models in practice. These tools can also be used to label predictions in real-time (as they are sometimes incorrect), which in turn improves the models' accuracy.

The process flow explained above uses BI knowledge, data insights and labeled data sets to produce trained models that produce real-time predictions - these models are returned to the BI platform when they are found to be accurate and robust. Evaluation of these models is done in an experimental manner, as will be further discussed in more detail in the case study that follows.

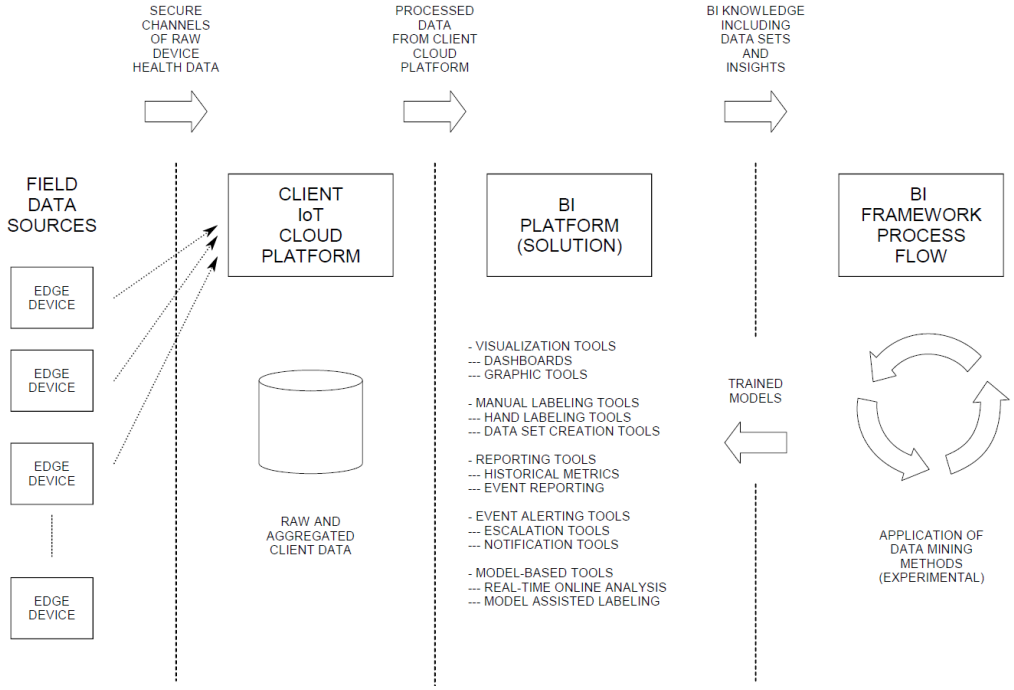

Figure 5.4: BI framework

5.2 BI Framework Case study

This section contains an implementation of the Development and Operational phases describe in Section 5.1.

In the Development phase, the current IoT system is evaluated and an overview of the important components and interfaces is provided. The requirements of the BI solution are also provided, solutions are identified and evaluated while considering all constraints, available resources, and TPMs. The BI platform is implemented and evaluated in the Operational phase to determine if requirements have been met and where improvements can be made. Implementation of threshold detection, KPI identification, and model design and all performance evaluation are done by means of experimentation.

5.2.1 Cellular IoT System

The main functionality of the IoT system under evaluation is to provide cellular communication services to clients. This utilizes a cellular IoT device combined with a data communication system (machine-to-machine messaging system) that implements all of the application requirements. An additional component is the remote management system that forms the data source to the BI platform.

5.2.1.1 Architecture

The relevant components of the system are illustrated in Figure 5.5. The system provides a secure data link between the client application devices and the client application services (show in blue). A local interface allows on premise maintenance, diagnostics and setup (shown in orange). Remote management capabilities are provided using a secure data link allowing certain health status indicators to be available for analysis (shown in green). The management function also provides remote configuration, software updates and remote diagnostics. The remote management data is grouped into separate accounts that group devices per client, which allows application-specific behaviour to be grouped per account.

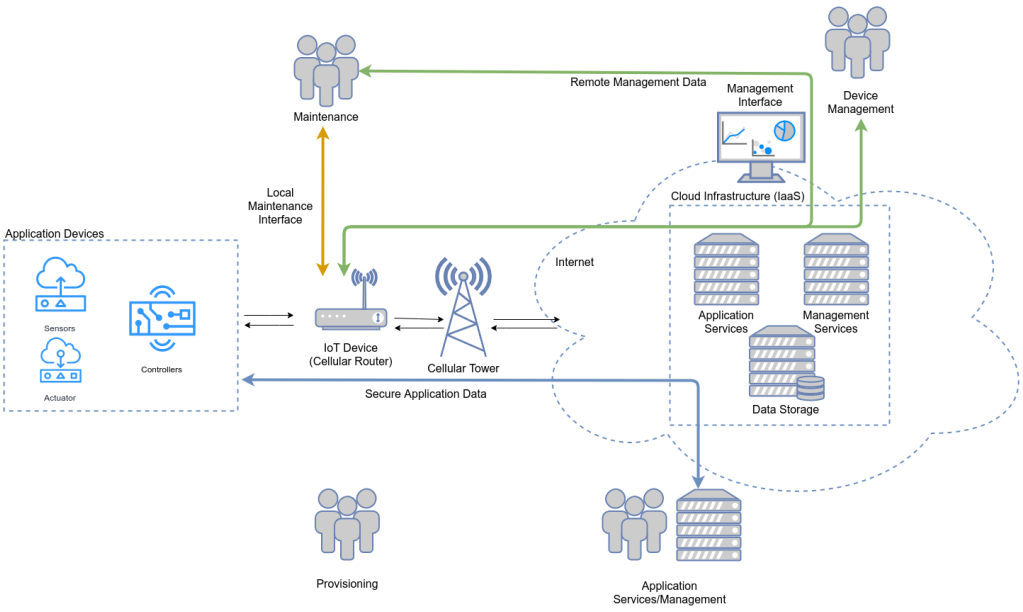

Figure 5.5: Cellular IoT Architecture

The main goal of the BI platform is to maximize up-time and to detect anomalous data that indicates faulty devices. These have to be diagnosed and repaired remotely, or have to replaced and repaired with field repairs.

The devices include a backup battery that extends the up-time of the device in case of a power failure, which is a common occurrence in developing countries. The devices support two cellular communication Active Network Providers (ANPs), increasing the availability.

The remote management system contains components and resources that can be used as elements of the BI platform. This will form the basis of the BI platform in that the remote management system can provide data, processing power and storage, and can take action upon detection of actionable events.

5.2.1.2 Data Sources

The main data source available for use in the BI platform is data created on the remote devices and communicated to the remote maintenance services. This data is processed and saved in a document-oriented database. The data can be divided into two main categories, namely (i) event-based data and (ii) health status data. The full status of the device is communicated at regular intervals as well as when a specific event occurs, these events include when a device changes ANP, changes power status, or when an automatic reboot occurs. The status contains multiple different numerical and binary type fields. Each of these fields is saved as a time series, called a metric, as described in more detail below. The binary data is accumulated as a count for a specific time frame, the numerical data is saved as the total of a specific event, while the number of occurrences is used to determine averages, where applicable.

The time series is aggregated into three separate collections containing hourly documents with minute-wise granularity, daily documents with hour-wise granularity, and monthly documents with daily-wise granularity. The different collections improve query times of database queries that require calculated totals over a larger time period. The documents in the different collections all use similar data and structure, where a document is comprised of the following information:

- A unique identifier;
- Metadata relating to the time, metric type, account and device;

- An object containing the specific time instance (minute, hour, day) and the total and number of samples for that time instance;
- A summary of the total, minimum, maximum and number of samples for that hour, day or month.

Table 5.1 provides a summary of all of the available metrics saved for each device. The aggregation type indicates the storage method of the metric, and the different types are explained below:

- Average indicates that each update calculates a total and increases an occurrence count. The average should be used as the value for that time frame;
- Count indicates that the number of occurrences of a specific event for the relevant time frame is present in the data;
- Delta type indicates that the value contains the total for the time frame only as a delta increase. This allows only communicating a changes in data from a device.

Table 5.1: Available data summary (per device)

Metric Name	Data Type	Aggregation Type	Description
Mains Voltage	Millivolt (mV)	Average	This is the measured input voltage for the device.
Battery Voltage	Millivolt (mV)	Average	This is the measured battery voltage of the device.
RSSI	RSSI	Average	This is the measured Received Signal Strength Indicator (ANP Only).
BER	BER	Average	This is the measured bit error rate (ANP Only).
Link Quality	Percentage	Average	This is a calculated percentage of the link quality of the device based on a range of different parameters. This is calculated on the device
Auto Reboots	Count	Count	This is the count of auto reboot occurrences for the specific time frame.
ANP Swaps	Count	Count	This is the count of ANP swap occurrences for the specific time frame.
Data Usage	Bytes	Delta	This is the total amount of data usage for the specific time frame.

The above data was available from the remote management system as obtained from deployed units in the field.

5.2.2 BI Requirements

The requirements for the BI platform are extracted by addressing short term and long term goals, considering constraints, and evaluating resources.

The functional requirements (F/R) were defined and divided into three main categories mentioned in Section 5.1.2.1.2.

5.2.2.1 Data Requirements

The data management requirements relate to the storage, access and manipulation of the data to provide data to the rest of the components of the BI platform.

- Data storage (F/R 1.1) - The data must be stored in a scalable, cost effective database. This requirement was already met by the current system with a document-oriented database;
- Data processing (F/R 1.2) - The raw data from the devices must be parsed, formatted and saved. This requirement was already met by the current system with a JavaScript server environment combined with a database driver allowing access to the database;
- Data aggregation (F/R 1.3) - Grouping data over different time frames, aggregating as a total or an average had to be done. This was possible using the database aggregation functionalities;
- Data transformation and extraction (F/R 1.4) - Converting data into a relational form that can be used in other BI tools had to be done. Secure access to the data was possible in a relevant format;
- Data labeling (F/R 1.5) - Labeling specific events and anomalies had to be done using a user friendly interface that allows different business users to classify events and label anomalies;
- Data cleaning (F/R 1.6) - Removing missing and unreliable data from the database or after extraction was possible.

5.2.2.2 Analytics Requirements

Data analytics requirements relate to the tools and techniques required to create models that allow detecting or predicting specific events or anomalies. This includes different statistical and machine learning techniques and methods. These tools can consist of SaaS platforms that provided an interface to achieve these tasks.

- Time series analysis (F/R 2.1) - Time series analysis and predictions was required to evaluate time series anomalies;
- Machine learning techniques (F/R 2.2) - Different machine learning techniques were required to achieve clustering and classification of the data;

- Business rules (F/R 2.3) - Customizable business rules on specific KPIs that create automatic email alerts were required;
- Feature extraction (F/R 2.4) - Converting features into required formats as well as creating new features from existing features had to be done.

5.2.2.3 Visualization Requirements

- Time series graphs (F/R 3.1) - Visualizing time series data for different metrics and units. This requirement was already met using a dynamic web application combined with a charting library;
- Drill down capabilities (F/R 3.2) - Intuitively drill down had to be done into data to discover the cause for observed events;
- Customizable dashboards (F/R 3.3) - Customizable dashboards that indicate KPIs had to be provided;
- Reports (F/R 3.4) - Customizable PDF reports that use historical data to summarize KPIs must be provided;
- Event alerting (F/R 3.5) - Email alerts for specific business rules an anomalous events must be implemented.

5.2.3 BI Solution

Implementing the BI platform consisted of evaluating different solutions for each of the required components. These components where identified by evaluating the requirements, constraints and available resources.

The different components could be defined as functional units that interface with one another to provide the final BI platform solution, and to address the requirements. As the management system described in Section 5.2.1.1 already contained components that met some of the requirements, these were expanded to create the final solution. This provided additional resources that could be used to address the remaining requirements.

Figure 5.6 shows all of the functional units that were integrated to form the final BI platform. The following is a summary of the different functional units. Functional units 1.1, 1.2, 3.1 and 3.2 are included in the current management system and resolves relevant requirements shown in Table 5.4 in green. These functional units were reused as an available resource to also satisfy other requirements. The current solution is described as follows:

- Document-oriented Database (F/U 1.1) - This is the database used to store all of the management data and is used to store all of the additional data generated by the ML models, labeled data and any additional KPIs identified in the Operational phase. The database also includes an aggregation pipeline that is used to aggregate the data over different time frames to allow extraction of relevant KPIs (elaborated upon in Section 5.3);
- JavaScript server environment - This is the main server that processes the data generated by the devices. This server interfaces with the database to allow access to the data using a RESTfull API and a dynamic web application. The server also interfaces with a mailer package (F/U 3.4) that allows alerts when specific events occur, as well as with the report generating module (F/U 3.3) to enable historical reports.
- Dynamic web application (F/U 3.1) - This is the main interface that is used on browser-enabled devices. The web application provides a platform that communicates with the server via a RESTfull API to securely access all relevant data. This data is then displayed using F/U 3.2 as time series charts with drill-down capabilities or as different customizable dashboards that display KPIs. This also provides an interface to label data, request reports and configure business rules and thresholds for alerts.
- Charting library (F/U 3.2) - This is used to create different types of graphical visualizations of the data. The library also allows interactions with the charts enabling selection, magnification (zooming) and drill-down capabilities. This is also used to provide a user friendly interface to select data points and label relevant points as anomalous.
- Reporting library (F/U 3.3) - This provides the ability to generate PDF reports of the main KPIs summarized over a historical period. An interface is created using F/U 3.1 where different report types can be generated using time and device filters to create a custom report. The report is generated using F/U 1.1 and F/U 1.2 to collect the relevant data and format the data to allow generating a PDF report.

- Email alerting service (F/U 3.4) - All anomalies that are detected can be configured to generate email alerts. These anomalies can be detected by applying business rules, or by thresholds that have been exceeded by KPIs that are evaluated (for easily detectable anomalies), or anomalies detected by more complex models that perform anomaly detection or classification.

The functional unit F/U 2 contains all of the tools used to analyze data, clean data, transform data and to generate models that can detect anomalies, classify events etc. One function of these tools is to develop and deploy models that make predictions. Predictions can be included as a new source of data that is ingested by the system, this is defined as F/U 2.3.

The following contains an evaluation of possible solutions evaluated considering four different criteria, namely (i) the skills required to use the platform, (ii) the features included and the complexity of the output models, (iii) the deployment options, and (iv) the costs and effort of development and deployment. These technical performance measures of each solution are ranked and an importance weight is used to score each platform accordingly.

Three platforms were evaluated, as follows:

- Python - This contains developing models and scripts in house using available libraries like *pandas* [55], *scikit-learn* [56] and *statsmodels* [57];
- BigML - This is a machine learning platform that incorporates a range of different features, including an online model machine learning dashboard that can be used to load and transform data, train different machine learning models and deploy these models.
- Microsoft Azure Machine Learning Studio -

  > "Azure Machine Learning Studio is a GUI-based integrated development environment for constructing and operationalizing Machine Learning workflow on Azure."

Table 5.2 indicates the trade-off between these solutions. It can be seen from the scoring results, taking all these parameters into account, that Python is the preferred solution. The is primarily due to the cost impact of implementing production models on the other two platforms. A free tier is available that allows experimenting with the models available on these platforms, but implementation for production purposes is costly. A hybrid approach is

implemented that utilizes the BigML platform to experiment and evaluate different models, after which implementation of the best preforming model was done using the Python programming language using the libraries listed above.

Table 5.2: Data analytics trade-off study

		Python		BigML		Azure	
TPM	TPM Weight	Score	WS	Score	WS	Score	WS
Skill Requirement	0.5	2	1	8	4	6	3
Feature Complexity	0.8	10	8	8	6.4	7	5.6
Deployment Options	0.7	5	3.5	6	4.2	8	5.6
Overall Cost	0.9	10	9	4	3.6	2	1.8
Total Score		21.5		18.2		16	

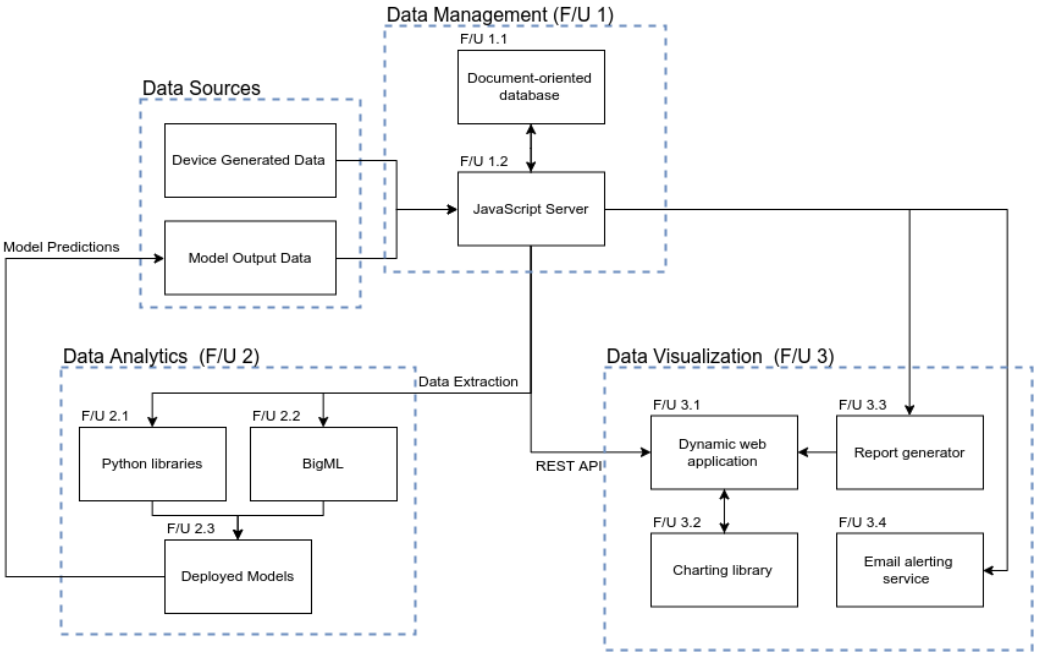

Figure 5.6: Functional Units

Table 5.3 shows a list of functional requirements and their reference numbers.

Table 5.3: Requirement summary

Requirement reference	Description
F/R 1.1	Data storage
F/R 1.2	Data processing
F/R 1.3	Data aggregation
F/R 1.4	Data transformation and extraction
F/R 1.5	Data labeling
F/R 1.6	Data cleaning
F/R 2.1	Time series analysis
F/R 2.2	Machine learning techniques
F/R 2.3	Business rules
F/R 2.4	Feature extraction
F/R 3.1	Time series graphs
F/R 3.2	Drill down capabilities
F/R 3.3	Customizable dashboards
F/R 3.4	Reports
F/R 3.5	Event alerting

Table 5.4 shows the allocation of functional requirements to solutions, in summary.

Table 5.4: Requirements allocation

		Functional Units								
		F/U 1.1	F/U 1.2	F/U 2.1	F/U 2.2	F/U 2.3	F/U 3.1	F/U 3.2	F/U 3.3	F/U 3.4
Functional Requirements	F/R 1.1		X							
	F/R 1.2	X	X							
	F/R 1.3	X	X							
	F/R 1.4	X	X				X			
	F/R 1.5	X	X				X			
	F/R 1.6		X	X	X					
	F/R 2.1			X	X	X				
	F/R 2.2			X	X	X				
	F/R 2.3	X	X							
	F/R 2.4		X	X	X					
	F/R 3.1	X	X				X	X		
	F/R 3.2	X	X				X	X		
	F/R 3.3	X	X				X	X		
	F/R 3.4	X	X						X	
	F/R 3.5	X	X							X

5.3 Implementation using Experiments

This section describes tasks that followed the BI framework process flow model described in Section 5.1.2.2. Each experiment relates to a task in the process flow that produces business knowledge, business rules, KPIs or models that can detect or classify specific events. All of the experiments use data from a single client account and can be conducted on the data of another account, should this be a future requirement.

The experiments follow the BI framework process flow model explained in Figure 5.3, experiment 1 implements P/R 2, 3 and 4 while the other experiments implement P/R 1, 3 and 5.

5.3.1 Experiment 1

5.3.1.1 Purpose

This experiment evaluates data to identify KPIs and determines if there are business rules or thresholds that can be used to alert when these KPIs indicate irregular behaviour. The thresholds provide a simplified way to detect anomalies and can be set up by the business user using a graphical interface. This also generates simple models that provide a baseline performance.

5.3.1.2 Method

The method involves combining business knowledge with data understanding by means of visually exploring the data to determine if certain KPIs contain anomalies that can be detected simply by using thresholds. This utilizes the charts created on the BI platform to visually inspect the data to determine if clear anomalies can be seen. The data is aggregated across an account and evaluated to determine a relevant threshold. Each account contains a different number of units and thus the threshold is account specific. The effectiveness of the threshold can be evaluated by labeling a subset of the data as anomalous and comparing the actual labeled data to the values predicted as anomalous (from using this threshold). The evaluation process includes determining the confusion matrix and calculating the F_β-score with β set to 3 to prioritize a recall. All data was hand labeled by the researcher to ensure data integrity.

5.3.1.3 Data

The following charts provide a visual representation of account-wide hourly metrics that contain anomalies. Thresholds were implemented to detect anomalies, which are indicated with a red bar label. This allows one to visually evaluate the effectiveness of the thresholds. These charts can be used to establish a viable threshold that can be used to indicate an anomaly. This threshold can then be used to evaluate new metric data points and alert on when a value for a specific hour exceeds the threshold.

Figure 5.7 shows the number of reboots across an entire account grouped into hourly bins. Anomalies are shown in red, the figure shows two large point anomalies that, after further investigation, indicated all of the devices preformed a single reboot, this usually relates to software updates or user requested actions. The reboot threshold was set up to alert when the reboot count across the specific account exceeded 50 times in one hour. The data set contains 12465 data points each representing one hour, all of these points were hand-labeled by the researcher to ensure integrity of the data.

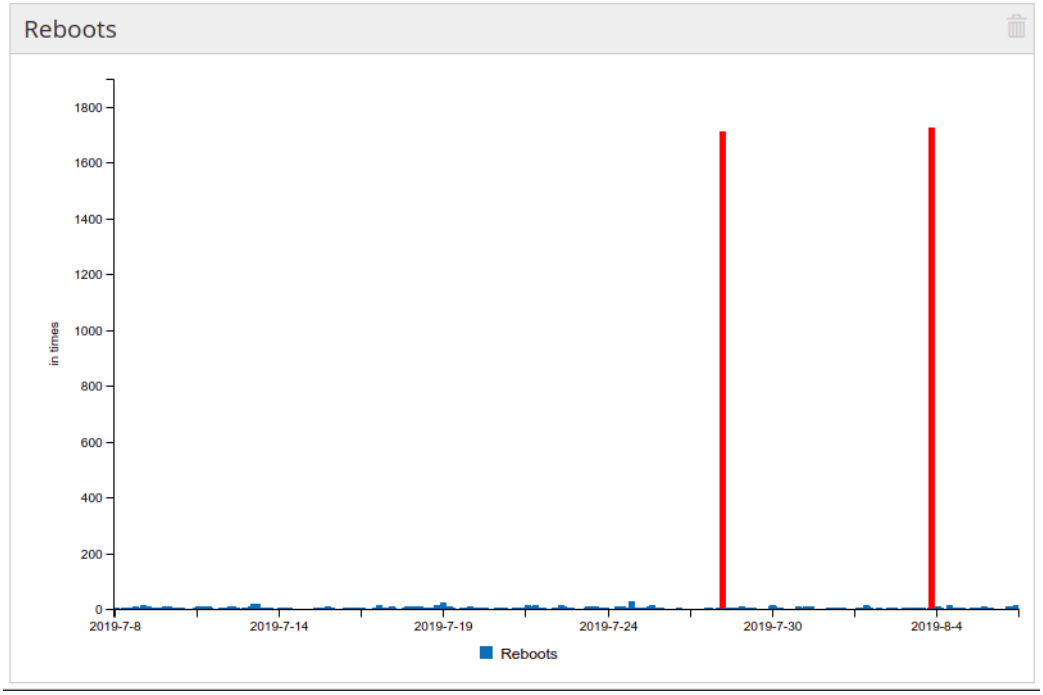

Figure 5.7: Hourly reboot count (Account wide, anomalies in red)

Figure 5.8 shows the total amount of data usage for all devices in a specific account, grouped into hourly bins. The data usage threshold was set up to alert when the data usage amount across the specific account exceeded 1GB in one hour. The data set contains 14196 data points each representing one hour. As before, all data was hand-labeled to ensure training and test data integrity.

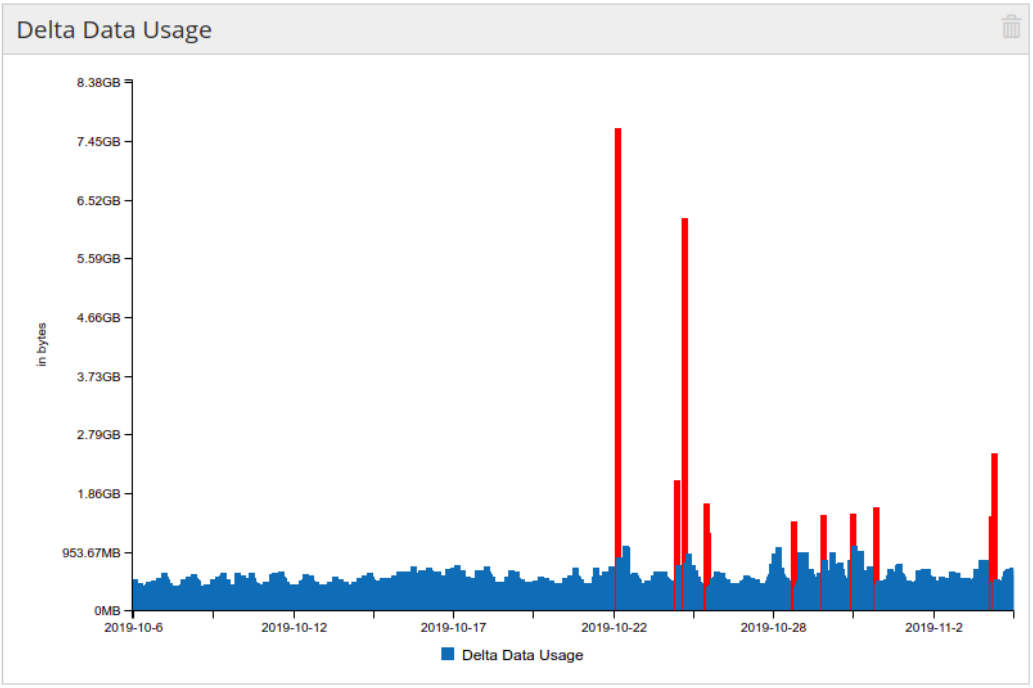

Figure 5.8: Hourly data usage (Account wide, anomalies in red)

5.3.1.4 Results

5.3.1.4.1 Reboots

Table 5.5 shows the confusion matrix for the reboot anomalies as obtained for the test data (previously unseen data). It can be seen that a total of 327 anomalies were predicted from the 12465 samples and 58 of these predictions were correct while 10 anomalies were not identified. These models do not include training (user generated threshold) and the entire data set is used for testing.

The F_3-score is calculated as 0.62.

Table 5.5: Reboots - Threshold model confusion matrix

n=12465	Predicted: Normal	Predicted: Anomaly	
Actual: Normal	12128 (TN)	269 (FP)	12397
Actual: Anomaly	10 (FN)	58 (TP)	68
	12138	327	

5.3.1.4.2 Data Usage

Table 5.6 shows the confusion matrix for the data usage anomalies. It can be seen that a total of 297 anomalies were predicted from the 14196 samples and 62 of these predictions were correct while 16 anomalies were missed.

The F_3-score is calculated as 0.62.

Table 5.6: Data usage - Threshold model confusion matrix

n=14196	Predicted: Normal	Predicted: Anomaly	
Actual: Normal	13883 (TN)	235 (FP)	14118
Actual: Anomaly	16 (FN)	62 (TP)	78
	13889	297	

5.3.1.5 Analysis of results

The results indicate that a simple threshold model can be very useful to implement as an initial time series anomaly detection technique. This model is efficient, but requires business knowledge and visual data exploration to determine a valid threshold. This technique fails when the time series contains trends and seasonality that exceeds thresholds.

The accuracy of these models is acceptable for an initial implementation considering the low performance cost and relative ease of implementation.

5.3.2 Experiment 2

5.3.2.1 Purpose

This experiment compares and evaluates different unsupervised time series anomaly detection techniques. The experiment and technique is documented in detail in Appendix B, this provides a summary of the methods and results. Three methods were evaluated namely a SARIMA model, a LSTM model and a support vector machine (SVM) model. In this experiment the LSTM model is improved and compared to the SARIMA model while the SVM model is excluded due to undesirable results. The hourly metrics are used to evaluate the models but can be modified to use any of the granularities and metrics available. Data was hand-labeled by the research to ensure data integrity.

5.3.2.2 Method

The anomaly detection method consists of predicting the time series and evaluating the difference between the prediction and the actual values. The two evaluated models have different training methods, namely (i) the SARIMA model that requires continuous training with a 7 day window and predicting 1 day, then shifting the training window one day, and (ii) the LSTM model that is trained with 1 year of data and used to predict the remainder of the data set. The LSTM can be retrained periodically but this was not included in the experiment. Hyper-parameter tuning is used to evaluate a range of different model parameters, this is explained and detailed in Appendix B. In this experiment, the best preforming models are evaluated as options. The LSTM is modified from the model used in Appendix B by including additional features extracted from the time series, including a moving average of the previous 12 hours and the hour of day.

To evaluate the actual performance of the model, the anomalous data was hand labeled to indicate the ground truth and compared to the predicted anomalies. An anomaly is determined by evaluating the difference between the actual and predicted values and using this difference to compute a windowed interquartile range (IQR). An anomaly is present when the difference is larger than the product of a specific threshold and the IQR. This allows calculating different F_β scores for different thresholds, thus indicating the performance of the model across multiple thresholds.

5.3.2.3 Data

The data evaluated consisted of the account wide ANP swaps and the account wide delta data usage. The ANP swap metric contains seasonality and a trend that disqualifies the simple threshold method, while the delta data usage is evaluated to determine if the accuracy can be improved when compared to the simple threshold method.

The data contains 14200 hourly data points relating to roughly a year and a half worth of hourly data. Figure 5.9 shows selected seasonal deviations and point anomalies contained in the ANP swaps metric (to demonstrate the data characteristics).

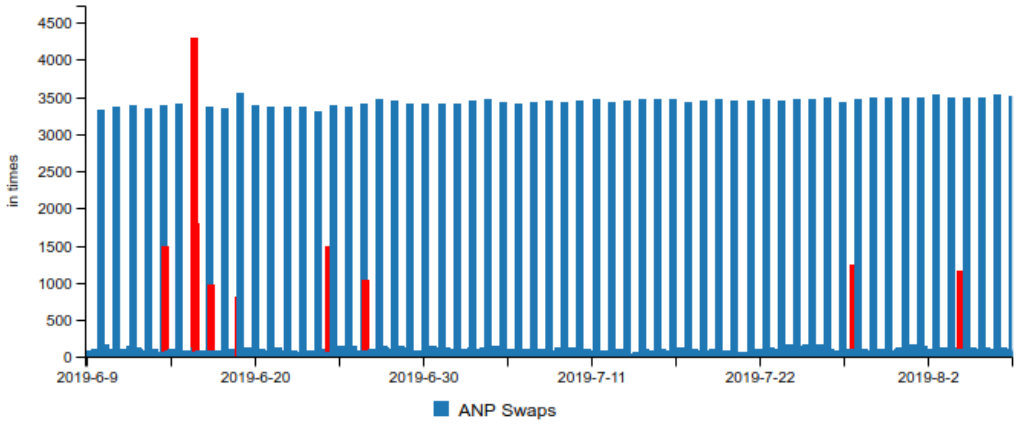

Figure 5.9: Hourly ANP Swaps (Account wide, anomalies in red)

5.3.2.4 Results

The best preforming parameter for each model was discovered using hyper-parameter tuning techniques described in Appendix B. The SARIMA and LSTM models were evaluated using a F_β-score with β equal to 3. Table 5.7 shows the highest F_3-score for each model when adjusting the threshold for predicting anomalies.

The Figures 5.10 to 5.17 show the confusion matrix for each of these models and data combinations.

Table 5.7: Time series anomaly detection model results

Metric	Model	F3-Score
ANP Swaps	SARIMA	0.95
ANP Swaps	LSTM	0.96
Data Usage	SARIMA	0.78
Data Usage	LSTM	0.90

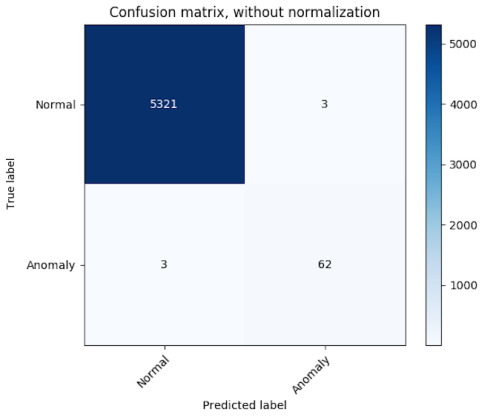

Figure 5.10: Non-normalized Confusion Matrix for ANP SARIMA model

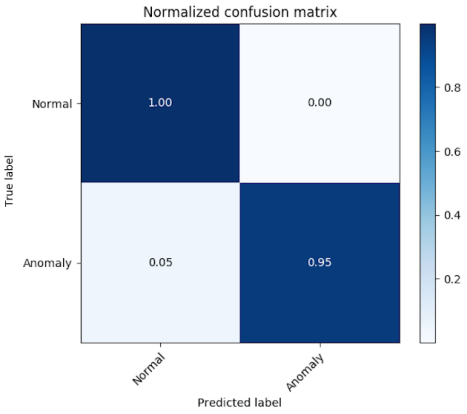

Figure 5.11: Normalized Confusion Matrix for ANP SARIMA model

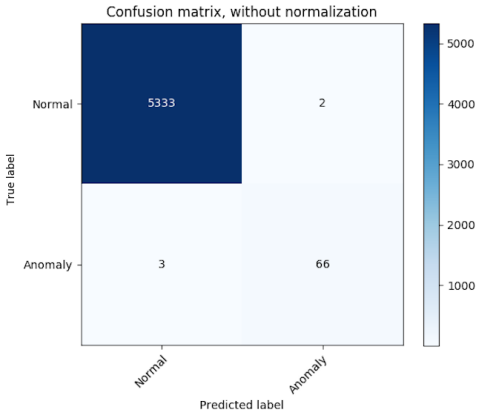

Figure 5.12: Non-normalized Confusion Matrix for ANP LSTM model

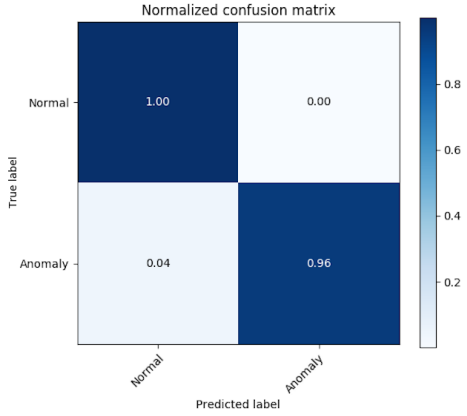

Figure 5.13: Normalized Confusion Matrix for ANP LSTM model

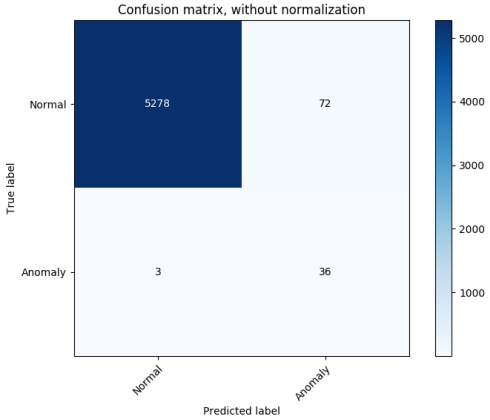

Figure 5.14: Non-normalized Confusion Matrix for data usage SARIMA model

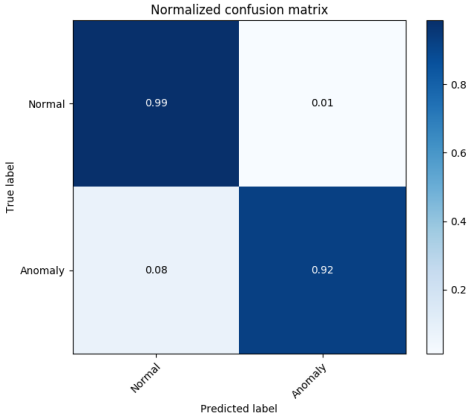

Figure 5.15: Normalized Confusion Matrix for data usage SARIMA model

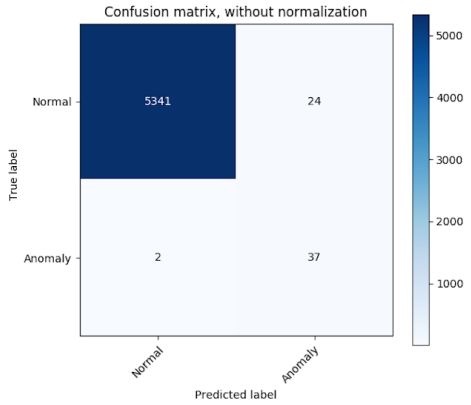

Figure 5.16: Non-normalized Confusion Matrix for data usage LSTM model

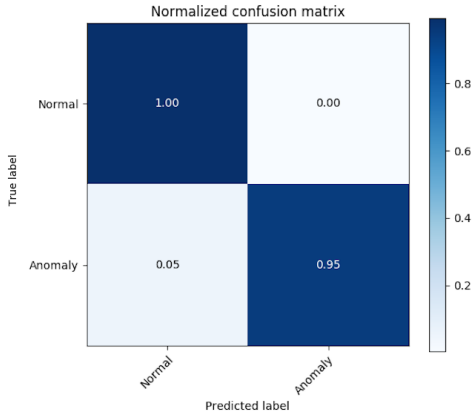

Figure 5.17: Normalized Confusion Matrix for data usage LSTM model

5.3.2.5 Analysis of results

The initial implementation of the LSTM models provided slightly worse results than the SARIMA model results found in Appendix B. After extracting additional features and presenting them to the LSTM model, the LSTM model performance improved significantly for the data usage data and slightly for the ANP data. The predictions of the LSTM model handles changes in the data over time better than the SARIMA model.

An advantage of the LSTM model in this environment is that the model does not need to be retrained as often as the SARIMA model.

5.3.3 Experiment 3

5.3.3.1 Purpose

The availability of field devices is a key performance measure of the system and it is important to maximize the availability of the devices by whatever means. When a device losses communication with the maintenance services for a specific period of time, a communication loss event can be created and saved. Identifying this event (from data) by means of classification provides crucial insights that can be used to improve the overall system availability.

From evaluating the data associated with these events, a variety of causes may be attributed to the loss of communication. Of the most common instances include devices that drained batteries normally (the rate of battery discharge was normal) and devices that indicated a "bad communication signal"before losing communication.

The purpose is thus to evaluate the data leading to a "communication loss"event to determine underlying reasons for failure, and to predict a potential (imminent) communication loss. This information may be used to reduce the risk of losing communication.

5.3.3.2 Method

This method consists of extracting a range of features (from the historic data) before a specific event, to implement feature reduction techniques to reduce the number of features, and to cluster the remaining features into classes. The data, events, and clusters have to be evaluated and labeled in order to determine the accuracy of the model.

The Mean Shift algorithm was used to cluster the data, combined with bandwidth estimation, to automatically determine the optimal number of clusters.

5.3.3.3 Data

The feature extraction consisted of calculating the mean, standard deviation and slope (were applicable) of the different metrics and extracting the final state of the device as features, this produces 40 different features. After removing entries containing missing values, a total of 4619 entries remained.

Feature selection techniques were used to reduce the feature number to 7 features (refer to Appendix B):

- Last battery voltage;
- Last mains voltage;
- Last power status;
- ANP swap count standard deviation over the previous 24 hours;
- Battery voltage slope over the previous 6 hours;
- Mains average over the previous 6 hours;
- Mains standard deviation over the past 6 hours;

5.3.3.4 Results

The clustering technique clustered the data into 18 clusters. A subset of the data was evaluated to determine the validity of the clusters and the accuracy of the model. This process indicated that the 18 clusters would be grouped into 7 unique classes shown in Table 5.8.

Table 5.8: Cluster description

Class Label	Description
A	No apparent reason
B	Battery drained normally
C	Battery drained irregularly
D	Excessive ANP swaps (Indicating communication issues)
E	Battery not charging or disconnected
F	Device under test
G	No battery or mains readings

Figure 5.18 shows the confusion matrix on test data for the evaluated clusters and Figure 5.19. The weighted F-score for the model predictions is calculated as 0.97.

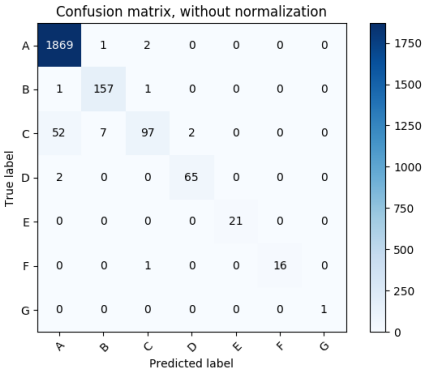

Figure 5.18: Clustering Confusion Matrix

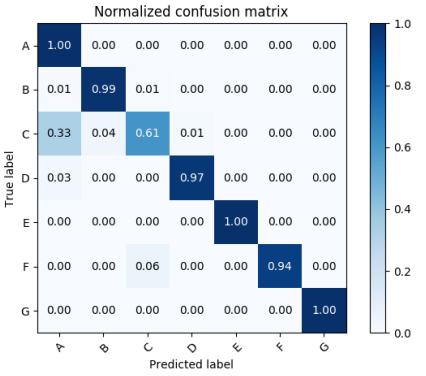

Figure 5.19: Normalized Clustering Confusion Matrix

5.3.3.5 Analysis of results

Clustering the data provided useful insights into the underlying causes of device communication loss. Classes C, D, E and G are considered anomalous and are flagged for evaluation by system operators.

The accuracy of clustering was evaluated and an F-score of 0.97 found to be acceptable. The resulting model was deployed into a production environment that detects communication loss events, extracts and formats relevant features, and predicts the communication loss cause.

This information is saved and can be included on a dashboard that indicates the most recent communication loss events and the predicted reasons, as well as the total number of communication loss events aggregated over a specified time, grouped per device. This view is shown in Figure 5.20 to indicate the 10

devices with the largest number of communication loss events over a specific 30 day period. It can be seen that the most frequently occurring event is the possible communication issues event, shown in blue, relating to class D. Further investigation revealed that the link quality of these devices are abnormally low, indicating that the cellular connection is unreliable. Device 8 indicated possible “bad battery”classifications. Using the interactive chart, the detailed device data can be further investigated. Figure 5.21 indicates a power loss and the battery that had drained abnormally fast. This can now be communicated through the system maintenance channel to address the undesirable device behaviour.

This process shows the value a clustering technique can add to the system when the data is provided to the correct system resources in the relevant format.

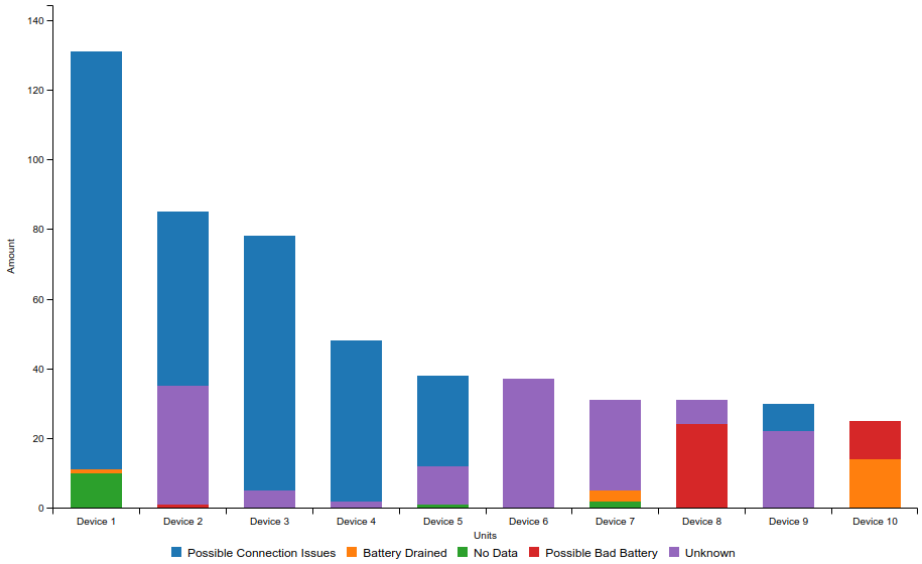

Figure 5.20: Aggregated communication loss events

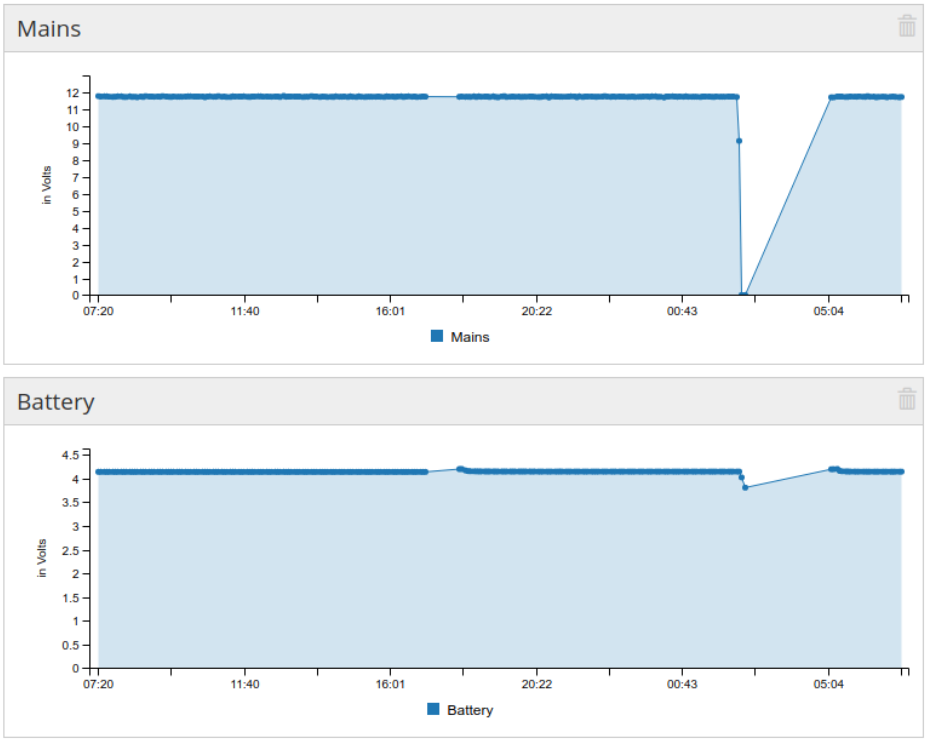

Figure 5.21: Detailed battery data

5.3.4 Experiment 4

5.3.4.1 Purpose

The purpose of this experiment is to investigate supervised classification techniques to improve the clustering model evaluated in experiment 3.

5.3.4.2 Method

The BigML platform was used to train and evaluate the supervised clustering models. The labeled data was extracted from the system, which included additional metrics not included in the clustering experiment. The data was cleaned and the remaining data was uploaded to the BigML platform.

The BigML platform includes an automated model scanning and parameter tuning process that evaluates a large number of different models and parameters to determine the best preforming model. These models include different regression models, decision tree models, neural network models etc. The platform supports ranking the models according to different performance metrics, the F_1-score is used to rank the models.

5.3.4.3 Data

The data used in this experiment is the hand-labeled data used in experiment 3, including additional metrics. The additional metric extraction resulted in additional missing values. After removing the entries with missing values, a data set with 707 entries remained.

The data was split into a training set used to train different models, and into a test set to evaluate the performance and identify the best performing model.

5.3.4.4 Results

The BigML OptiML process evaluated a total of 122 different models and model configurations of which 66 models were distinct models. Random decision forest and bootstrap decision forest were the top preforming models with the random decision forest having the best F-score of 0.74 on the training data.

The model was used to predict classes on the test set and used to create a confusion matrix. A final F-score was obtained for evaluation purposes. The F-score for the test set was calculated as 0.85, Figure 5.22 and 5.23 shows the confusion matrix and the normalized confusion matrix.

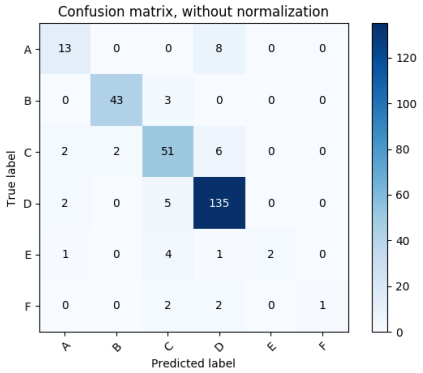

Figure 5.22: Supervised classification confusion matrix

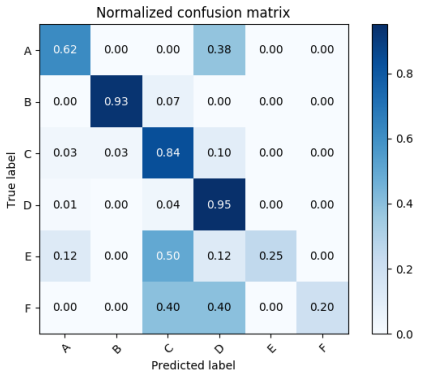

Figure 5.23: Supervised classification confusion matrix (normalized)

5.3.4.5 Analysis of results

It can be seen that not all of the classes were included in the test set, this is due to high class imbalance and a lack of sufficient labeled data. The model did not preform better than the clustering model. This is due to the issues mentioned above. The nature of anomalous classes are that these occur less frequently and thus additional techniques are required to address the high class imbalance.

The first obvious improvement is to obtain additional data that includes more instances of the anomalous classes. An additional approach could be to synthetically generate instances of these classes, this is called a synthetic minority over-sampling technique (SMOTE) [58]. Within the limited scope of this research, additional techniques were not employed and the LSTM model in Experiment 3 was deployed in practice.

5.3.5 Summary

The BI framework provided in this chapter addresses the need for a general framework with which to develop BI platforms for cellular IoT networks. The BI framework model effectively addresses the research solutions shown in Table 5.9:

Table 5.9: Solution Validation Matrix

		Research Solutions			
		Definition of BI and IoT Framework	Definition of system characteristics	Definition of intelligence framework	Integrated system perspective
Detailed Solutions	Artifact: Definition of BI framework model	↑		↑	↑
	Case study:				
	Detailed operational analysis	↑		↑	↑
	Implementation of the proposed BI framework for a Cellular IoT System	↑	↑		↑
	Extraction and evaluation of system characteristics (Experiments)		↑		↑
	Manual labeled data sets for system verification				↑
	IEEE conference proceeding				↑

Essentially, the BI framework comprises two phases, namely (i) a Development phase, and (ii) an Operations phase. During the Development phase, a platform is created on which data can be sourced, processed, analyzed and used to provide insights. In the Operations phase, the BI process flow model is used to define methods for extraction of intelligence.

In the real world application presented in this chapter, a solution was developed by applying the BI framework exactly as defined above. Cellular network health data was used as sources to the platform, including a number of specified KPIs and metrics. The data was then aggregated and provided for modeling purposes, as was done in 4 experiments. The first experiment used naive threshold techniques for anomaly detection, to good effect, but with limitations with respect to trends and seasonality present in time series data of this real-world nature. The second experiment compared unsupervised time series prediction methods, with the result that an LSTM network outperformed a SARIMA model both in terms of its performance and its robustness. The third experiment used classification principles (clustering) to find logical groupings of events, with the result that different classes were identified by a clustering network. Finally, in the last experiment the clusters were used to train models for classification, with BigML OptiML used to generate classifier models, with good results (but limited to an F-score of 0.85, which was still highly useful).

The fact that a process flow model was applied in a real-world project with good results validates the specific solutions to the concept solutions. A BI and IoT framework was developed and defined in the form of a model that can be applied to generate future solutions. System characteristics were defined for the cellular IoT network in the real world problem. The intelligence framework was defined specifically for the cellular network under investigation. Finally, the integrated solution was provided by applying the BI framework model to the real world problem. The platform was first created (as described) and the process flow model applied in a series of experiments.

The solution provided here thus address the initial real-world need, and conforms to the relevance requirement of the DSR paradigm. Also, as new knowledge was added by ADR, the requirements for ADR were also addressed. The QRM process was followed to validate the research.

Chapter 6

Validation and Conclusion

This research was conducted inside a DSR paradigm by following an ADR method, thus providing a solution to a real-world problem while generating additional knowledge in the process. The relevance cycle was used to extract different requirements and research challenges from the real-world need. These challenges were validated by experts' input to the system under evaluation, knowledge presented in literature and observations made from the real-world cellular IoT system.

From this research, an integrated BI framework is proposed that is utilized to implement a BI solution from a systems perspective. The following sections describe the research challenges and solutions, and explain how the artefacts created in this research validate the research solutions and corresponding research challenges.

6.1 Research challenges and solutions

6.1.1 Research challenge 1 - BI framework for cellular IoT network

1. Validation of research challenge: This primary research challenge is validated by a real world need for a BI solution that improves the effectiveness of an existing cellular IoT network. Expert input and observations from literature provide validation of the need for a BI framework. The

proposed BI framework integrates principles from systems engineering and the CRISP-DM processes to provide a BI framework process flow model and methods required to analyze, design and implement and operate a BI platform solution for a cellular IoT system.

2. Literature study: The research includes researching and describing different machine learning techniques used to extract insights from data. These techniques include supervised and unsupervised models to explore patterns and detect anomalies in different data types and structures.

3. Concept and specific solutions: The lack of a BI framework for cellular networks required definition of a process flow model that any developer can use to create a BI platform. The defined BI framework process flow model was used to create a BI platform solution to the real-world IoT problem. A detailed operational analysis was conducted, followed by implementation of the proposed framework process model by implementing the tasks of the flow model in a series of experiments.

6.1.2 Research challenge 2 - IoT system characteristics

- Validation of research challenge: The need to define an IoT network's system characteristics was observed from literature (mainly the lack of literature on cellular network characteristics). A study of Systems Engineering literature confirmed the need for performance measures as a means to indicate system behaviour. Observation on the IoT system under evaluation confirmed that system characteristics had not been defined at the onset of this research.

- Literature study: Literature on cellular communication theory provided valuable understanding into the architecture of a general cellular IoT system and the parameters and metrics available in such a system. Literature on data mining and the techniques and components associated with this discipline verified the need to generate techniques and models that are able to provide insights into the important system characteristics. Systems engineering literature confirmed the need for a process of technical performance measurement.

- Concept and specific solutions: The BI framework process flow model provided a means to knowledge discovery, which was used to determine KPIs and system behavioural characteristics. This process flow produced models as output by utilizing the CRISP-DM process. These models were evaluated and implemented using the BI framework process flow model and provided valuable predictions and detection of failure events.

6.1.3 Research challenge 3 - Intelligence ontology

- Validation of research challenge: The need for an ontology for cellular IoT systems was underscored by the lack of clear definition of an IoT system, as such. An ontology specific to cellular networks did not exist at the onset of this research, and it was necessary to provide such an ontology for the sake of alignment and addressing the correct requirements from the real world problem.
- Literature study: The IoT and BI literature topics contributed towards the definition of an ontology as terminology, processes, architectures and other definitions were obtained and applied to the IoT network problem.
- Concept and specific solutions: The definition of a BI framework model and a detailed operational analysis provided a general ontology on intelligence and the extraction of insights from data. This includes different architectures, processes, models, and terminology. From this, requirements could be defined and a process be followed to provide a BI platform. The combination of information was also used to provide a better understanding of the utilization of IoT network data and converting data into intelligence in the IoT environment.

6.1.4 Research challenge 4 - Integrated systems perspective

- Validation of research challenge: BI is often applied in isolation, and an integrated approach is not always followed. This may be due to lack of information from field devices, or simply a lack of understanding. This was observed at the onset of the study, namely that cellular modems

do not usually provide sufficient data from which to extract BI. The client system in this case was designed to be informative, and the system architecture and functions could be used to provide an integrated solution.

- Literature study: Literature on IoT, AI and machine learning, combined with the principles from Systems Engineering (specifically the principle of integration) resulted in the definition and implementation of an integrated solution. General challenges faced in implementing a BI system as described in literature were addressed in the implementation to ensure the same mistakes and pitfalls would be avoided.
- Concept and specific solutions: A BI framework model was developed from inputs obtained from literature, specifically by using principles of cellular networks (architectures), AI and machine learning theory as part of Data Mining, and Systems Engineering. Systems Engineering provided valuable input in the Development and Operational phases of the BI solution. The implementation of the BI solution is described in this dissertation as a case study and validates the proposed BI framework process flow model. The implementation phase consisted of analysing the IoT cellular network and resulted in an operational analysis which included the architecture of the system, the data sources available, and the different users and interfaces. The implementation produced a BI solution that could be used to extract insight from the available data sources in the Operational phase. This Operational phase included 4 experiments that produced insights that could be converted into business knowledge, KPIs and models. Evaluation of these models required hand labeling multiple data sets that formed part of the data sources. The process flow model was thus validated and the associated successful method resulting from the process (the LSTM model) could be deployed as a solution.

6.2 Contributions

This section summarizes the research contributions that validate the research solutions, which in turn validate the research challenges shown in table 6.1.

- BI framework for cellular IoT (model design) - This is the main artefact that integrates systems engineering concepts and data mining concepts to provide a BI framework for development of a BI platform solution in a cellular IoT environment - please refer to Figure 5.4 for a graphical representation of the BI framework. The definition of the framework is considered to be a design contribution, including the overall BI platform design process, the definition of a BI platform, and the definition of a BI framework process flow model;
- Implementation of a BI platform (case study) - This provides clear evidence of the implementation and operation of a BI solution for an existing cellular IoT system, also shown in Figure 5.4. All the tools shown in the diagram were developed by the researcher and used to conduct this design research. The case study includes smaller meta-artefacts that contribute to the KB, namely the operational analysis of the evaluated system that provided a model of the existing IoT network; experiments that provided output models and examples of data mining tasks which validated the Operational phase of the BI framework; and finally hand-labeled data sets (from the visualization and labeling tool set) are also included as contributions.
- IEEE conference publication - An IEEE Africon international conference article that focuses on the different experiments and data mining techniques used in this study as attached in Appendix B

Table 6.1: Validation Matrix

Chapters	Cloud based BI framework for a cellular IoT network					
3	Real world definition of need		↓			↓
3	Observations from Cellular Network applications			↓		↓
3	BI Publications		↓		↓	
	Information Sources	Research Challenges	Lack of implementation framework for BI in cellular IoT networks	System characteristics unknown	Lack of intelligence ontology	Lack of Integrated Application
	Focus Areas					
4.1	IOT		↑↓		↓	
4.2	BI		↑↓		↓	↓
4.3	Data Mining and Machine Learning		↓	↓		↓
4.4	Cellular Communication Systems			↓		
4.5	Systems Engineering		↓	↑↓		↓
	Focus Areas	Research Solutions	Definition of BI and IoT Framework	Definition of system characteristics	Definition of intelligence framework ontology	Integrated system perspective
	Detailed Solutions					
5.1	Artifact: Definition of BI framework model		↑		↑	↑
5.2 - 5.3	Case study:					
5.2.1	Detailed operational analysis		↑		↑	↑
5.2.3	Implementation of the proposed BI framework for a Cellular IoT System		↑	↑		↑
5.3	Extraction and evaluation of system characteristics (Experiments)			↑		↑
5.3	Manual labeled data sets for system verification					↑
Appx. B	IEEE conference proceeding					↑

6.3 Summary and future work

This research addressed the need for a BI platform for cellular IoT networks. The DSR paradigm was used in conjunction with the ADR methodology, and managed by QRM validation. Research challenges were derived, literature focus areas defined and a literature study conducted, research solutions were proposed and implemented, and finally, validation was provided by tracing the research and design process.

The artefact was developed first defining a BI framework model for development of future BI platforms. This model was applied to a real-world problem, namely the development of a BI platform for cellular IoT networks. The solution was validated when it was implemented on an existing, operational IoT network platform and successfully evaluated in a case study. Experimental results confirm that the performance of the AI models was acceptable and that the integrated solution addressed the initial need not only functionally, but also in terms of quality.

Bibliography

[1] A. R. Hevner, "A three cycle view of design science research," *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.

[2] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Quarterly*, vol. 37, no. 2, pp. 337–356, 2013.

[3] Lindgren, Sein, Henfridsson, Rossi, and Purao, "Action Design Research," *MIS Quarterly*, vol. 35, no. 1, p. 37, 2017.

[4] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. R. Daimlerchrysler, C. Shearer, and R. W. Daimlerchrysler, "Step-by-step data mining guide," *SPSS inc*, vol. 78, pp. 1–78, 2000. [Online]. Available: http://www.crisp-dm.org/CRISPWP-0800.pdf

[5] B. S. Blanchard and J. E. Blyler, *SYSTEM ENGINEERING MANAGEMENT*, 5th ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2007, vol. 41, no. 23.

[6] J. E. W. Holm and G. P. R. van der Merwe, "QUALITY RESEARCH MANAGEMENT IMPROVES DESIGN RESEARCH EFFECTIVENESS," *South African Journal of Industrial Engineering*, vol. 30, no. 3, pp. 238–252, 2019.

[7] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[8] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.

[9] V. Vaishnavi and B. Kuechler, "Design Science Research in Information Systems," *Ais*, p. 45, 2004. [Online]. Available: http://desrist.org/desrist/content/design-science-research-in-information-systems.pdfhttp://www.desrist.org/design-research-in-information-systems/

[10] J. Iivari, "Nothing is as Clear as Unclear," *Scandinavian Journal of Information Systems*, vol. 19, no. 3, pp. 111–117, 2007.

[11] M. Hung, "Leading the IoT," p. 29, 2017. [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

[12] R. D. McLeod, K. Ferens, and M. R. Friesen, "The IoT: Examples and trends," *Proceedings - 2015 International Conference on Computational Science and Computational Intelligence, CSCI 2015*, no. May 2016, pp. 336–339, 2016.

[13] S. Kiran and S. B. Sriramoju, "A study on the applications of IOT," *Indian Journal of Public Health Research and Development*, vol. 9, no. 11, pp. 1173–1175, 2018.

[14] P. Newman, "THE INTERNET OF THINGS 2019," Business Insider Intelligence, Tech. Rep., 2019.

[15] K. Al-Gumaei, K. Schuba, A. Friesen, S. Heymann, C. Pieper, F. Pethig, and S. Schriegel, "A Survey of Internet of Things and Big Data integrated Solutions for Industrie 4.0," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2018-Septe, pp. 1417–1424, 2018.

[16] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[17] A. Vakaloudis and C. O'Leary, "A framework for rapid integration of IoT Systems with industrial environments," *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 601–605, 2019.

[18] Q. Xiaocong and Z. Jidong, "Study on the structure of "Internet of Things(IOT)" business operation support platform," *International Conference on Communication Technology Proceedings, ICCT*, pp. 1068–1071, 2010.

[19] A. E. Hakim, "Internet of Things (IoT) System Architecture and Technologies," no. March, pp. 0–5, 2018.

[20] T. S. Rappaport, *Wireless Communications- Principles And Practice*, 2nd ed. Prentice Hall, 2002.

[21] J. D. Gibson, *The Mobile Communications Handbook*, 2nd ed. Springer, 1999.

[22] R. Mulcahy, "Business Intelligence Definition and Solutions," 2013.

[23] B. Evelson and M. Bennett, "The Forrester Wave™: Enterprise BI Platforms With Majority On-Premises Deployments, Q3 2017 A Significant Reshuffling Of Vendor Positions In A Changing Landscape Key takeaways," Tech. Rep., 2017.

[24] Better Buys, "The Definitive Guide do Business Intelligence," Better Buys, Tech. Rep., 2015. [Online]. Available: https://www.betterbuys.com/wp-content/uploads/2015/05/The-Definitive-Guide-to-Business-Intelligence.pdf

[25] T. Liyang, N. Zhiwei, W. Zhangjun, and W. Li, "A conceptual framework for business intelligence as a service (SaaS BI)," in *Proceedings - 4th International Conference on Intelligent Computation Technology and Automation, ICICTA 2011*, vol. 2, 2011, pp. 1025–1028.

[26] B. Ramesh and A. Ramakrishna, "Unified business intelligence ecosystem: A project management approach to address business intelligence challenges," in *PICMET 2018 - Portland International Conference on Management of Engineering and Technology: Managing Technological Entrepreneurship: The Engine for Economic Growth, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2018.

[27] J. Han, M. Kamber, and J. Pei, *Data Mining (Third Edition)*, 3rd ed., J. Han, M. Kamber, and J. Pei, Eds. Boston: Morgan Kaufmann, 2011.

[28] S. Angée, S. Lozano, E. Montoya-Munera, J. Ospina Arango, and M. Tabares, "Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-organization Big Data & Analytics Projects: 13th International Conference, KMO 2018, Žilina, Slovakia, August 6–10, 2018, Proceedings," 2018, pp. 613–624.

[29] S. Ben-David and S. Shalev-Shwartz, "Understanding Machine Learning: From Theory to Algorithms," Tech. Rep., 2014. [Online]. Available: http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

[30] A. Habeeb, "Artificial intelligence Ahmed Habeeb University of Mansoura," *Research Gate*, vol. 7, no. 2, 2017.

[31] P. Joshi, *Artificial Intelligence with Python*. Birmingham: Packt, 2017.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning*. [Online]. Available: http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop-PatternRecognitionAndMachineLearning-Springer2006.pdf

[33] P. Montague, "Reinforcement Learning: An Introduction, by Sutton, R.S. and Barto, A.G." *Trends in Cognitive Sciences*, vol. 3, no. 9, p. 360, 1999.

[34] R. Adhikari and R. K. Agrawal, "An Introductory Study on Time Series Modeling and Forecasting," Tech. Rep.

[35] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[36] M. Bodén, "A Guide to Recurrent Neural Networks and Backpropagation," 2001.

[37] X. H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of Long Short-Term Memory (LSTM) neural network for flood forecasting," *Water (Switzerland)*, vol. 11, no. 7, 2019.

[38] Y. Heryadi and H. L. H. S. Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM," in *2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings*, vol. 2017-November. Institute of Electrical and Electronics Engineers Inc., 3 2018, pp. 84–89.

[39] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. September, pp. 1–58, 2009.

[40] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017. [Online]. Available: https://ac.els-cdn.com/S0925231217309864/1-s2.0-S0925231217309864-main.pdf?_tid=f1a71094-fc86-11e7-85c0-00000aab0f02&acdnat=1516304397_f4336a3f6b93e63e311e51f0272bf741

[41] M. A. Prado-Romero and A. Gago-Alonso, "Detecting contextual collective anomalies at a Glance," *Proceedings - International Conference on Pattern Recognition*, vol. 0, no. April, pp. 2532–2537, 2016.

[42] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust Random Cut Forest Based Anomaly Detection On Streams," *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, 2016. [Online]. Available: https://d1.awsstatic.com/whitepapers/kinesis-anomaly-detection-on-streaming-data.pdfhttp://jmlr.org/proceedings/papers/v48/guha16.pdf

[43] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[44] Y. Lei, "Network Anomaly Traffic Detection Algorithm Based on SVM," in *Proceedings - 2017 International Conference on Robots and Intelligent System, ICRIS 2017*. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 217–220.

[45] P. Joshi, *Artificial Intelligence with Python*. Birmingham: Packt, 2017.

[46] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, 2015.

[47] Y. Sasaki and R. Fellow, "The truth of the F-measure," Tech. Rep., 2007. [Online]. Available: https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf

[48] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006. [Online]. Available: http://www.forecasters.org/data/m3comp/m3comp.htm

[49] A. Botchkarev, "Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology," 2018. [Online]. Available: http://arxiv.org/abs/1809.03006

[50] ——, "Evaluating Performance of Regression Machine Learning Models Using Multiple Error Metrics in Azure Machine Learning Studio," *SSRN Electronic Journal*, no. January 2018, 2018.

[51] W. Wang and Y. Lu, "Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model," *IOP Conference Series: Materials Science and Engineering*, vol. 324, no. 1, 2018.

[52] US Department of Defense Systems Management College, "US Department of Defense Systems Management College," *22060-5565*, no. January, p. 222, 2001. [Online]. Available: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf

[53] NASA and E. J. D. Hoffman, *NASA Systems Engineering Handbook 1995*, 1995, no. June. [Online]. Available: www.sti.nasa.govhttp://adsabs.harvard.edu/full/1995NASSP6105.....S

[54] Incose, "The Guide to the Systems Engineering Body of Knowledge (SEBoK)," *Insight*, vol. 5, no. September, pp. 36–39, 2011. [Online]. Available: http://g2sebok.incose.org/app/mss/menu/index.cfmhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+Guide+to+the+Systems+Engineering+Body+of+Knowledge+(SEBoK)#0

[55] W. McKinney, "pandas: a Foundational Python Library for Data Analysis and Statistics."

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 2825–2830, 2011. [Online]. Available: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[57] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57, no. Scipy, 2010, p. 61.

[58] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Tech. Rep., 2002.

Appendix A

General IoT architecture

BI Framework

Via Humans

Automated

Control via Intelligence

Intelligence

AI and Analytics - Business Intelligence Extraction

Intelligence

Control via Intelligence

LONG-TERM REACTION

Control via Information

Information

AI and Analytics - Management Information Extraction

Information

Control via Information

MEDIUM / LONG-TERM REACTION

Big Data

Presentation of Information

Database of Operational Data

Processing of Data

Database of Event Data

SHORT / LONG TERM REACTION

IMMEDIATE / SHORT-TERM REACTION

Control of "Things"

Executive Workforce

Control via Humans

Internet

Management Workforce

Operational Workforce

Sensor

GSM / LTE-M

NETWORK OF INTEREST

Cellular Tower

GSM Link

GSM Link

GSM Link

Proprietary Network

Control

Monitoring

SOME SYSTEMS MONITOR AND CONTROL LOCALLY AND ONLY COMMUNICATE ON EXCEPTIONS

Low Data Rate Local Wireless Network

Lora WAN Sigfox

Sigfox has its own back-haul network into the Internet

IoT Controller

Sensor

IoT Controller

Sensor

High Data Rate Local Wireless Network

Local Wireless Network

Router

Sensor

IoT Controller

IoT Controller

Sensor

Appendix B

IEEE conference publication

Cloud-Based Business Intelligence for a Cellular IoT Network

1st Johann Erich Wolfgang Holm
School of Electrical, Electronic and Computer Engineering
North West University
Potchefstroom, South Africa
johann.holm@nwu.ac.za

2nd Leon William Moolman
School of Electrical, Electronic and Computer Engineering
North West University
Potchefstroom, South Africa
liaanmoolman@gmail.com

3rd Gabriël Petrus Rossouw van der Merwe
Managing Director
Jericho Systems Pty Ltd
Potchefstroom, South Africa
rossouw@jerichosystems.co.za

***Abstract*—This paper presents a cloud-based business intelligence (BI) implementation for a cellular Internet of Things (IoT) network. A Design Science Research (DSR) paradigm, combined with elaborated Action Design Research (eADR) was used to ensure a workable artifact is delivered. The real-world problem is that, in the cellular network considered here, network health status was not initially visible in an intelligent and actionable way. The network health status is used to ensure service availability and includes different health indicators, of which measurements are made at regular intervals. Not all IoT edge devices have health indicators available, but the network under evaluation provided sufficient data from which to extract anomalies. Experiments were conducted to identify the most appropriate anomaly detection technique from three options, namely SARIMA, SVM and LSTM techniques. Anomalies were linked to system operational failures, in turn to be addressed by appropriate standard operating procedures of a larger maintenance system. Finally, a clustering algorithm was evaluated for automated recognition of anomalous events, showing that anomalies may be clustered in a useful way using the Mean-Shift clustering algorithm, and also identifying additional health indicators that support anomaly classification.**

***Index Terms*—Business intelligence, Internet of Things, anomaly detection, clustering, health status**

I. Introduction

This paper evaluates three different anomaly detection techniques and a clustering algorithm aimed at improving an Internet of Things (IoT) network's business intelligence (BI).

The IoT network under consideration utilizes two different network providers. Improving device reliability and system health reduces maintenance costs and increases availability. The main research challenge is to improve overall system availability by detecting anomalies that cause reduced performance and availability. This requires identification of relevant system health indicators and investigating different anomaly detection techniques. DSR and eADR were used to guide this research and design a workable solution to the problem.

The first focus is on time series anomaly detection techniques, namely Seasonal Auto Regressive Integrated Moving Average (SARIMA), Long Short-Term Memory (LSTM) and a one-class Support Vector Machine (SVM). The best of these will be used to detect meaningful changes in network health status. In addition, relevant health status indicators were identified for real-world implementation. An experiment was conducted to support this effort.

The second focus is on clustering, so as to group data into meaningful classes that can be actioned in a field service workflow system. Clusters are evaluated to determine if they provide classes relevant to anomalous operational events (in this case, failures). A second experiment using the Mean-Shift algorithm was conducted to find additional relevant health indicators.

Finally, a conclusion is presented to show that anomaly detection and classification are useful to automate detection of failures in the larger real-world IoT network system.

II. Cellular IoT system ontology

Fig. 1 shows the IoT network system. This system comprises three domains of interest. The first domain is an application domain that contains a secure data link between application devices and application services. The second domain comprises physical provisioning and maintenance. The third domain is a remote monitoring and management system.

The application is used to provide cellular communication services to clients. This is done using a cellular IoT device (called a router for sake of simplicity) and improving the reliability of the router will improve the availability of the communication link. A remote management system monitors and ensures availability of routers. If faulty routers can be detected automatically, those devices can be physically replaced by creating tickets for field repair. The remote management system thus plays an important role in the effectiveness (availability) of the IoT system.

Effectiveness of the management system can be increased using artificial intelligence to provide real-world actionable intelligence. This is done by providing failure events and decision support information to system operators, allowing these operators to make informed decisions that lead to resolution of faults. This requires different system components that can be integrated into a business intelligence solution.

Ramesh [1] suggests that the main reason for BI projects failing is due to companies not treating BI as a continuously evolving system element that aligns business operations with business goals. Some of the more important components of a

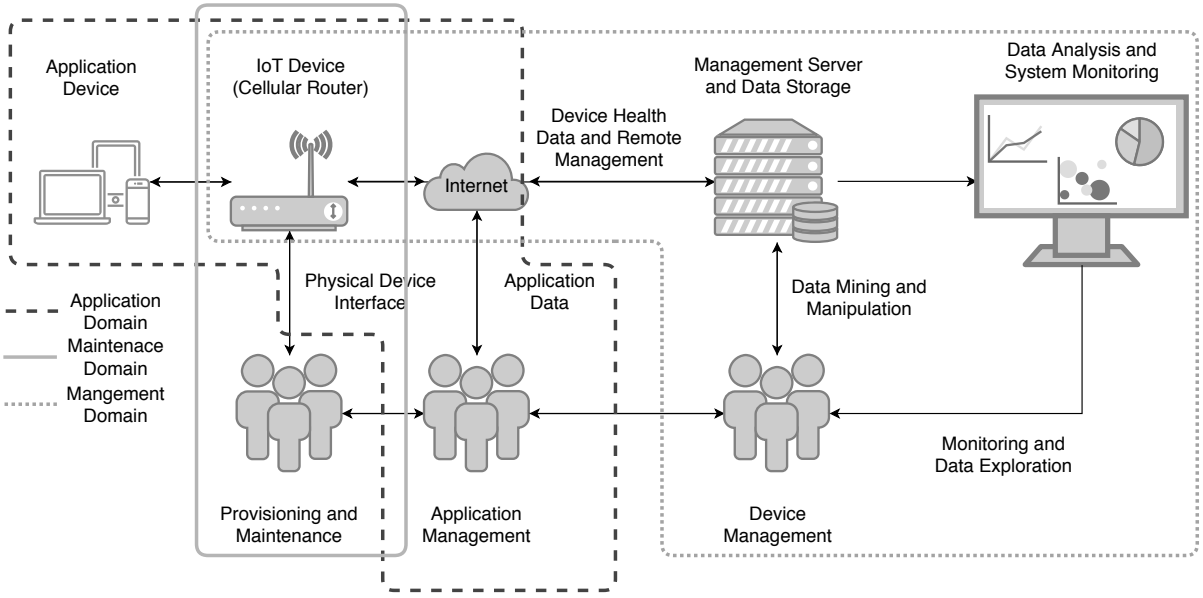

Fig. 1. Cellular IoT System Diagram.

BI solution include different levels of reporting, *ad hoc* queries and data drill down, dashboards, alerts and different data analysis techniques [2]. The first step in implementing a BI solution is to determine information and intelligence required to identify and manage exceptions, for which BI components have to be selected. The second step is to determine a road map and set long term and short term goals in terms of BI components as deliverables. These goals should act as guides, as a BI solution should ideally be dynamic and adaptable as knowledge of the system characteristics increases.

After initial drill-down, and subsequent reporting capabilities have been implemented, key performance indicators (KPIs) may be identified by evaluating events of interest and their underlying causes. Different analytic techniques for detecting or predicting these events should be evaluated by correlating causes and effects that indicate exception events. Basic statistical and aggregation techniques, (averages over different time frames, min/max values of specific features over different devices) should provide a good initial analysis. These techniques should increase in complexity (anomaly detection, classification and clustering, predictions) until an acceptable and beneficial solution has been identified. The solution can then be implemented and evaluated.

Finally, it is important to understand that a BI solution is at its core a decision support system that allows operators to explore and evaluate data in a flexible and supportive way [3].

III. Health indicators

For the IoT network under investigation, a detailed log of all measurements is recorded on the router. It is however not sensible to save this amount of detailed information. Thus, key events and features are regularly communicated to a cloud server and stored in a central database.

The routers contain backup batteries that provide additional up-time when a power failure occurs, as is common in developing countries and in cases where unwanted access to the power plug is possible. Routers also support two cellular communication Active Network Providers (ANPs) to improve availability by selecting the ANP that is available at a specific point in time and defaulting to a preset ANP when available.

An event is generated and saved when a device:

- Changes ANP;
- Changes power status;
- Loses communication for a predetermined time period;
- Reboots;

The device has the following indicators that are saved at specific intervals:

- Data usage;
- Battery voltage;
- Mains power voltage;
- Bit Error Rate (BER);
- Received Signal Strength Indicator (RSSI);
- Number of ANP swaps;
- Number of auto reboots;
- Link quality (computed on the device).

The following indicators are aggregated into account wide features:

- Delta data usage (incremental increase in usage);
- Number of ANP swaps;
- Number of reboots.

A. Anomalies

An anomaly can be seen as data behavior that differs from a well defined normal pattern of behavior [4]. Anomalies often indicate critical actionable information. This section discusses different anomalies that have been detected in this network.

The anomalies discussed can be subdivided into two main categories based on the data used to detect these anomalies. The first category consists of time series data that may contain seasonality and trend. The second category consists of extracted features from time series data at relevant time stamps.

There are 3 main types of time series anomalies that occur in this system. The first is a point anomaly, the second is a trend change and the third is a seasonality deviation.

The following elaborates on the different time series anomalies. Account-wide features provide a very good overview of the network system, thus anomalies in these features usually indicate that an event occurred that affected multiple devices. The metric containing the number of ANP swaps has seasonal spikes at expected time intervals due to pre-programmed swap times. The absence of these spikes usually indicate server issues. If a spike occurs outside of this expected pattern, it usually indicates device updates that cause an ANP swap on reboot or it can also indicate an ANP outage.

Account-wide reboots are relatively limited with seasonality. If a point anomaly occurs, it usually indicates device updates (when new configurations or firmware versions are loaded). When a level change occurs, it usually indicates that a single device is faulty and causing an unusual number of reboots.

The hourly delta data usage is usually very consistent with a strong seasonal component. Point anomalies usually indicate a single unit with an anomaly relating to high data usage. A level change could indicate that devices have changed data usage behavior and should be investigated.

B. Data

Data used in this investigation is time series data comprised of health indicators and status measurements as described above.

Health indicator values are saved periodically while a specific device is online. This is also saved if an event cause a significant change. The data is then aggregated into monthly, daily, hourly and minute bins. This is done to increase the effectiveness of extracting features over a range of time. The data is stored in a NoSQL document styled database.

The data used in the experiments ranges from the start of 2018 up until April 2019.

The total number of ANP swaps, total number of reboots and delta data usages are also aggregated into an account wide metric. This allows account wide anomalies to be detected using a single time series.

The health status collection contains a maximum of 1 year of data. This collection contains roughly 200 million documents.

The hourly metrics collection is the largest collection and contains roughly 600 million documents.

C. Anomaly detection methods

Two main data analytics methods were evaluated. The first is time series anomaly detection techniques and the second is clustering techniques grouping units into normal and anomalous classes.

1) Time Series Anomaly Detection: Anomaly detection identifies unusual data behavior that could indicate problematic devices or events as a first failure indicator without in-depth system knowledge.

Different time series anomaly detection techniques exist [4]. One common technique is predicting the time series and evaluating the difference between the actual and predicted values. Another technique is to remove seasonality and trend from the time series and determine if the remaining behavior can be considered anomalous.

- An ARIMA model is a combination of three different models, namely *Auto-Regression* (AR), *Integrated* (I) and *Moving Average* (MA). An extension to the ARIMA model is a Seasonal ARIMA (SARIMA) model. The SARIMA model can be represented as follows:

$$SARIMA(p,d,q) \times (P,D,Q)^{\mathrm{m}} \tag{1}$$

  The p and P parameters indicate non-seasonal and seasonal AR order. The d and D parameters indicate non-seasonal and seasonal I order and the q and Q parameters indicate non-seasonal and seasonal MA order. The m parameter indicates the seasonal period or lag that the seasonal AR, I and MA models use [5].
- Long short-term memory (LSTM) is a recurrent neural network (RNN) variant that incorporates long range dependencies and effective learning of sequences with varying lengths. This is done by using three gates in each LSTM unit. Input gates decide which inputs should be remembered, a forget gate decides when a value should be forgotten and an output gate decides if a value should be contained in the output [6].
- *Support Vector Machines* (SVM) are supervised machine learning techniques that can be used for regression and classification. Classification using SVMs is done by using selected specific support vectors to define a hyper-plane that separates different classes [7].

2) Clustering: Clustering is an unsupervised machine learning technique that is most commonly used as a data exploration technique. This is due to the fact that unlabeled data is used in clustering and the ground truth is unknown. This implies no real success evaluation process can be done without labeling data. Two different clustering techniques can produce very different results while still being valid clusters [8]. Thus, using a clustering technique to explore clusters and using this as support for labeling data is useful. Many different clustering techniques exist, some require the number of clusters to be known and others use techniques to estimate the number of clusters. Mean Shift is a technique where bandwidth estimation can be used to determine the optimal number of clusters, or classes [9].

3) Model Evaluation: Evaluating models is essential in machine learning. Due to the fact that most machine learning models and solutions can have multiple configurations or parameters, evaluating the performance of these different configurations and models provides the most appropriate solution.

Common techniques use the specificity and sensitivity to evaluate a model using a Receiver Operating Characteristics (ROC) plot. Saito *et al.* [10] indicated that the ROC curve can be misleading if the data set is unbalanced and alternative performance methods should be used. Precision-Recall Curves (PRC) provide a better indication for unbalanced data sets. This is due to the fact that the baseline model changes as class distribution changes. As anomalies are per definition less frequent than normal occurrences, the precision and recall is preferred as a performance method. The F_β-score is a combination of the precision and recall and defined as:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} (0 \leq \beta \leq \infty) \quad (2)$$

The β parameter controls the weight between P and R. A β larger than one ensures that recall has a stronger influence on the score [11].

IV. Research methodology

A. Design Science Research and Action Design Research

The research conducted in this study uses Design Science Research (DSR) as a research paradigm to focus on creating an artifact that solves a real world problem. This is done in three main cycles. The *Relevance Cycle* connects the design process with the application environment by providing opportunities or problems in the form of requirements. This cycle also provides a method of evaluating the artifact by testing in the application domain. The *Rigor Cycle* provides the connection between a knowledge base and the design process. This is done by preforming rigorous research to support the design process and to extract knowledge to put back into the knowledge base. The *Design Cycle* is the iterative process of evaluating inputs from the *Relevance Cycle*, extracting methods and theories to support the design from the *Rigor Cycle*, designing and implementing an artifact, testing the artifact using the *Relevance Cycle* against the requirements and adding to the knowledge base using the *Relevance Cycle* [12].

In this research, the real world problem is the need for a BI solution for an IoT network that provides a better view of systems health status. The identification of system characteristics that relate to anomalies relevant to the system health is the focus. Evaluation of different analytic techniques to detect or predict these anomalies is the main focus of experiments that were conducted as part of the design process.

Action Design Research (ADR) is a method for conducting design research where knowledge is extracted while conducting designs - this is done in four main steps [13]:

- Problem formulation;
- Building, intervention and evaluation;
- Reflection and learning;
- Formalization of learning.

ADR is thus a method that can be used in the DSR paradigm to deliver an artifact as part of research. Elaborated ADR (eADR) expands on the ADR methodology by subdividing the Problem Formulation step into a Problem Diagnosing and Concept Design steps. eADR also suggests that intervention and evaluation occur at every step and not just at the Building step [14].

V. Experimental Research

Experiments were conducted as the research part of the Action Design process with the purpose of identifying relevant system health indicators. Experiments done in this research comprise of feature extraction and selection, anomaly detection and clustering. These experiments focus on two main problems, namely (i) time series anomaly detection and (ii) unsupervised data exploration and clustering with the focus on classifying device events into normal events and anomalous events.

A. Experiment 1

1) Purpose: The purpose of this experiment is to compare different time series anomaly detection techniques based on performance metrics. It also briefly describes the process followed to select and pre-process features that were used.

2) Method: Each technique will be evaluated using the most complex time series feature, ANP swap count, grouped into hourly time bins. Anomalies in the features have been labeled for model evaluation purposes.

Three different time series anomaly detection techniques will be evaluated. The first is an SARIMA model, the second is a Long short-term memory (LSTM) recurrent neural network and the third is a one-class support vector machine (SVM) anomaly detection algorithm.

An auto SARIMA model is used on a sub set of each feature to determine the type of SARIMA model that should be used. The SARIMA model will then be repeatedly trained on a window of the data, predict 24 hours and determine if the actual values are anomalous. Due to the fact that the SARIMA model is sensitive to anomalies in the training data, anomalies will be removed once the model has detected those anomalies correctly. The LSTM model is trained with anomalies included in the data and without. The LSTM model is trained with 3000 and 5000 rows of data extracted into 24 features, each feature representing an hour and the label representing the next hour that should be predicted. Thus, a single row contains 24 features and one label. The remaining data is used for testing. The SVM model is trained on a training set where the seasonality and trend have been removed by using moving averages. The data is split into two testing sets and a training set. The first is used for hyper parameter tuning to select the best parameters for the SVM model using the F-score as the optimization metric. The second is used for evaluating the model.

The first two models predict the time series and if the series deviates from the prediction, it is labeled as an anomaly. The models provide a prediction which is then subtracted from the actual value at that time frame. A windowed interquartile range (IQR) [15] is calculated on the difference and an anomaly is determined if the difference is larger than a specified threshold multiplied by the IQR.

The SVM model provides an anomaly score and a threshold can also be set to determine if the specified value is an anomaly.

These thresholds can be adjusted to increase or decrease the precision, recall and F-score of the models. This is calculated using the confusion matrix for each threshold [10].

A single feature will be used to determine the best preforming model. The other features will then be evaluated with the best preforming model. The feature containing the number of ANP swaps has the most types and frequency of anomalies.

3) Data: The first step is to select relevant features for prediction. It was seen that the account wide aggregation of device features provided a good summary of the overall device health. Investigating these features indicated that the most anomalies could be seen in the hourly time granularity. The total number of ANP swaps, delta data usage and total number of reboots showed anomalies. These three features are used in hourly time bins. The data ranges from April 2018 until April 2019 and each feature contains 8942 rows or hours.

The data can contain varying seasonality and trend. The most common anomalies are point anomalies and seasonal deviation events. Some level changes also occur.

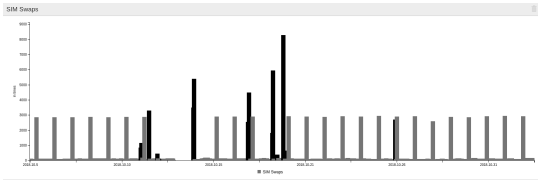

Fig. 2. Hourly ANP swap count

Fig. 2 shows a month of the feature containing the number of ANP swaps. The black bars indicate anomalies. It can be seen that there is a seasonal spike each day at the same time, when this spike is absent it could indicate server-related issues. The other anomalies shown could indicate an ANP outage or device updates.

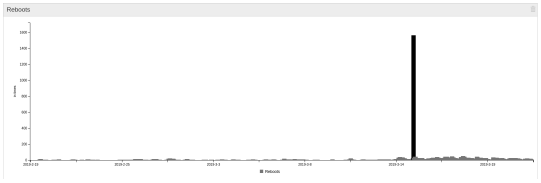

Fig. 3. Hourly Reboot Count

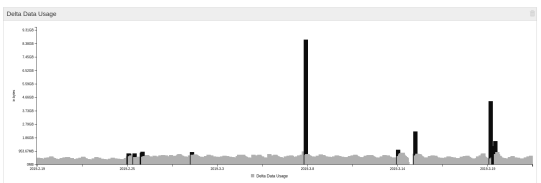

Fig. 4. Hourly Delta Data Usage

Fig.3 and Fig.4 shows a month of total number of reboots and delta data usage aggregated across an account. The anomalies present in these features, shown as black bars, are mostly point anomalies.

4) Results: Each model has specific parameters that can be adjusted to improve the results.

SARIMA model selection: Due to specific feature properties, specific SARIMA model orders may outperform others. The auto SARIMA algorithm uses a differencing test to determine the optimal non-seasonal and seasonal differencing order. The other parameters are determined by optimizing one of the common information criteria. After the SARIMA model orders have been determined, the train data window length can also be adjusted. Decreasing this window size can increase the ability of the model to adjust to changes in the data, but can reduce the overall performance of the model.

The optimal orders for the SARIMA model obtained for the number of ANP swaps are as follows:

$$SARIMA(4,1,1) \times (0,1,1)24 \tag{3}$$

The SARIMA model was evaluated with a 7 day, 10 day and 14 day window.

LSTM model selection: An LSTM network can be implemented in multiple different configurations. The LSTM used is a stacked LSTM to allow for increased model complexity [6]. Data containing anomalies and data with removed anomalies are used. It was observed that the LSTM provided better results with data without anomalies. Thus, further parameter tuning was done on this data set. The following list gives the different parameters used:

- LSTM 1 : 3000 training samples, original data, 100 hidden layers, 200 epochs;
- LSTM 2 : 3000 training samples, cleaned data, 100 hidden layers, 200 epochs;
- LSTM 3 : 5000 training samples, cleaned data, 100 hidden layers, 200 epochs;
- LSTM 4 : 3000 training samples, cleaned data, 200 hidden layers, 200 epochs;
- LSTM 5 : 3000 training samples, cleaned data, 300 hidden layers, 200 epochs.

SVM model selection: An SVM model is optimized by implementing a hyper parameter tuning technique. This is done by sweeping two parameters on specified values and optimizing for the F-score. Table I shows three models with its parameters. Each model and parameter pair can be further optimized by adjusting the anomaly threshold. The recall is an indication of the number of correctly labeled anomalies, while precision relates to the amount of noise that needs to be evaluated with the anomalies. A higher precision indicates a lower amount of noise. The F_β-score is a combination of the precision and the recall. As the recall has a higher importance for this application a β of 3 is used.

5) Analysis of Results: It can be seen that the SARIMA models perform best as it was least sensitive to adjustment in threshold whilst producing acceptable results overall. Due to

TABLE I
MODEL PERFORMANCE RESULTS

Model	*Threshold*	*Precision*	*Recall*	*F-score*
SARIMA 7-Day Window	2.0	0.31	0.95	0.79
SARIMA 7-Day Window	3.0	0.39	0.92	0.81
SARIMA 7-Day Window	4.0	0.45	0.89	0.81
SARIMA 7-Day Window	5.0	0.49	0.86	0.80
SARIMA 10-Day Window	2.0	0.29	0.93	0.76
SARIMA 10-Day Window	3.0	0.38	0.91	0.80
SARIMA 10-Day Window	4.0	0.43	0.86	0.78
SARIMA 10-Day Window	5.0	0.49	0.84	0.78
SARIMA 14-Day Window	2.0	0.29	0.93	0.76
SARIMA 14-Day Window	3.0	0.37	0.91	0.79
SARIMA 14-Day Window	4.0	0.41	0.88	0.79
SARIMA 14-Day Window	5.0	0.48	0.86	0.80
SVM Anomaly Detector	0.31	0.20	0.79	0.61
SVM Anomaly Detector	0.33	0.23	0.68	0.57
SVM Anomaly Detector	0.36	0.21	0.5	0.44
LSTM 1	2.0	0.08	0.74	0.41
LSTM 1	3.0	0.09	0.68	0.41
LSTM 2	2.0	0.06	0.78	0.36
LSTM 2	3.0	0.07	0.71	0.37
LSTM 3	2.0	0.09	0.64	0.40
LSTM 3	3.0	0.13	0.59	0.44
LSTM 4	2.0	0.24	0.95	0.73
LSTM 4	3.0	0.33	0.94	0.79
LSTM 4	4.0	0.38	0.92	0.81
LSTM 4	5.0	0.41	0.85	0.77
LSTM 5	2.0	0.11	0.82	0.50
LSTM 5	3.0	0.15	0.77	0.54

the fact that this is an anomaly detection system and that the number of anomalies is relatively low, it is important that the recall be high. If a minimum recall of 90% is required, the best model is the SARIMA with a 7 day training window and a threshold of 3.6. This produced a 90% recall and a 43% precision.

Following a similar process, SARIMA models were created for the total number of reboots and the delta data usage features.

The reboots feature used the following SARIMA model:

$$SARIMA(1,1,1) \times (0,1,1)24 \tag{4}$$

Using a threshold of 8 the recall was found to be 94% and the precision was 62% with an F-score of 0.89.

The delta data usage feature used the following SARIMA model:

$$SARIMA(2,1,3) \times (0,1,1)24 \tag{5}$$

Using a threshold of 5, the recall was found to be 83% and the precision was 19% with an F-score of 0.62.

The SARIMA model is the least complex to implement as there exists multiple libraries in different coding languages that allow implementing the models.

Although the SARIMA models performed well, there are optimizations that can be done on LSTM models, but suffice to say for the tests conducted here, the SARIMA models displayed the highest robustness. Data characteristics are heavily dependent on underlying processes and events, and not all anomalies can be labeled correctly as additional business knowledge is often required. The link between data science and business domain specialty is thus very important. It is important to label data correctly (using domain specialty knowledge) to improve overall model performance, and the SARIMA models will be adjusted and improved over time as more business knowledge emerges in the spirit of DSR.

B. Experiment 2

1) Purpose: The purpose of this experiment is to evaluate the feasibility of clustering techniques to indicate different reasons for device communication loss and to evaluate communication loss events that can be considered anomalous. Devices lose connectivity for a variety of reasons. Of these, signal loss, normal battery drainage and irregular battery drainage are examples. A clustering algorithm should ideally indicate if these classes are relevant and highlight additional classes that might be unknown.

2) Method: Features that correlate to classes of failures are unknown at the onset of the experiment. Thus, a wide range of different features are extracted by calculating averages and standard deviations of all available features over different time windows before an offline event occurs. This data is compared to features extracted from units that have been online for an extended period of time. Feature selection techniques are used to reduce the number of features to a smaller number. This improves the computational performance and potentially the accuracy of the model.

After the features have been identified, a clustering algorithm is selected by evaluating algorithm performance.

Due to the fact that the number of clusters in the data set is unknown, the Mean Shift algorithm is chosen as it can estimate the number of clusters. Evaluating clustering algorithm performance can be difficult as the ground truth is not always known. Thus, the clustering algorithm can be used to provide an indication of different classes. Identified samples can then be labeled according to classes as the ground truth, and classification algorithms can be used to improve the results of the clustering algorithm if the clustering performance is not acceptable.

3) Data: Clustering algorithms require a set of features and a time series needs to be processed before it can be used in the clustering algorithm.

Feature extraction was done by firstly identifying occurrences where a device had lost connectivity for more than 30 minutes and occurrences where a device had been connected for a long period of time. This was used as examples of offline and online labels, respectively. All of the time series features (mentioned above) were extracted as averages (avg) and standard deviations (SD) over a period of 6 hours, 12 hours and 24 hours. The discharge rate of a battery (over 6 hours) was also calculated using simple linear regression. The power status, last mains voltage and last battery voltage were also used as features.

In total, 40 different features were extracted and a binary output label for each was assigned. The data consisted of 22400 records. Some data could not be extracted as devices had not been online for an extended period before the event,

hence a loss in data. These rows were cleaned out and 17393 rows remained. Of these rows, 12774 rows were extracted from devices that operated normally and had not lost communication. The remaining 4619 rows contained features extracted from the time before a device had lost communication.

The best 15 features were selected using univariate statistical tests, and were used in clustering. The algorithm suggested 35 different clusters. This was overly specific and the number of features was reduced. The clustering algorithm also provided a score for each feature and the top 7 features were selected. The features resulting from clustering are:

- Last battery voltage;
- Last mains voltage;
- Last power status;
- Mains average over 6 hours;
- Mains standard deviation over 6 hours;
- Battery slope for last 6 hours;
- ANP swap count standard deviation over 24 hours.

4) Results: The clustering on the main features above provided 18 clusters. A subset of 2295 instances out of the total 4619 were evaluated and labeled. The labeling process indicated 6 unique classes. Table II describes all of the different classes identified by analyzing the clustering results. Some of the clusters contained slightly different data but could still be grouped in a single class. This is indicated in the table.

TABLE II
CLUSTERS AND RELEVANT CLASSES

Class Label	*Class Description*	*Cluster Labels*
A	No clear reason	0
B	Battery drained normally	1,7,8,11,18
C	Battery drained too fast	2,10,17
D	Excessive ANP swaps	3
E	Battery not charging	4,9
F	Device under test	5,6,12,13,14,16
G	No battery or mains readings	6

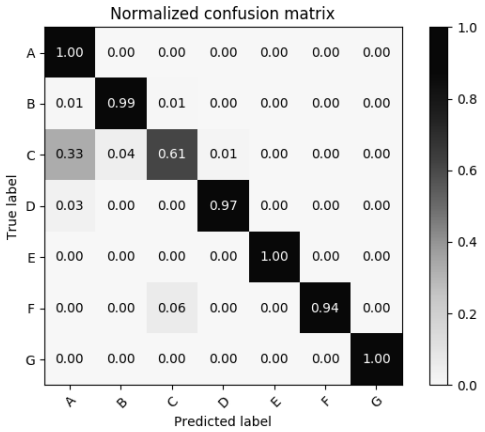

Fig. 5. Normalized Clustering Confusion Matrix

Fig. 5 shows the normalized confusion matrix and fig. 6 shows the confusion matrix without normalization. The overall

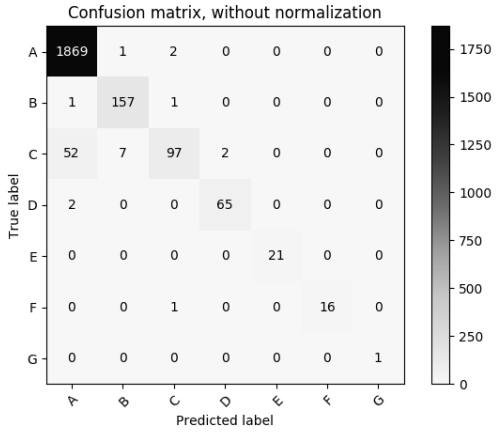

Fig. 6. Clustering Confusion Matrix

F-score of the model result using a weighted approach to account for class imbalance is 0.97.

5) Analysis of Results: The model provided acceptable results and indicated classes that had been unknown before the experiment commenced, namely classes E and G. The overall F-score of 0.97 is acceptable and does not need to be further improved for an initial model. Class A had the most occurrences and further investigation may be done at a later stage to find root cause events. Classes C, D, E and G are considered anomalous and must be flagged for investigation by system operators.

It can be seen that class C had the worst results, this is caused by devices that lost connectivity almost immediately after losing power. The clustering model labeled some instances as belonging to class A - as mentioned, this may be improved by using additional classification methods if necessary.

Sub-classes were identified while labeling the data - these sub-classes can be added to the data and a classification model can be used to improve usability of the results. These sub-classes mainly indicate different levels of battery health. At this point in the research, the clustering was found sufficient for operational support as confirmed by the client.

It is understood that the fault identification models will be implemented on a monitoring system and displayed on a stack for workflow purposes. Higher levels of intelligence may be extracted, for example component failures, manufacturing failures, or sabotage. A practical method may be to display communication loss on a near real-time dashboard for action by field service personnel.

VI. CONCLUSION

This paper presented research on business intelligence for cellular Internet of Things to ultimately improve system availability. To this effect, a Design Science Research approach was followed to address the real-world problem, namely to detect system anomalies relevant to communication failures.

Each field device reports its health status in an "always on" manner, with different status indicators available to be used

as features in an anomaly detection engine. The relevance of features had to be investigated using time series analyses as well as clustering techniques. This was done using two experiments that used actual field data as obtained from communication devices (routers).

In the first experiment, time series analyses were done and three different models were compared, namely SARIMA, LSTM and SVM. It was found that the SARIMA models outperformed the other models both in terms of robustness and model performance. The SARIMA model displayed the least sensitivity with respect to model parameters, whilst providing a recall score of above 0.9 and F-scores of higher than 0.8. Its closest competitor was the LSTM model with a recall score above 0.5 and F-scores of higher than 0.3.

In a second experiment, clustering was used to find relevant features and associated anomalies from field data, with anomalies clustered into classes. This experiment showed that it was possible to classify averages and standard deviations (lower order moments) of data prior to failures into relevant failure mode classes. The Mean Shift clustering algorithm was used to good effect in this experiment, and clusters were formed using health status indicator statistics. In total, 18 clusters were formed and reduced to 6 main classes. Two of these classes were considered as normal operational modes and contained 88% of the evaluated data. The remaining 12% consisted of 5 classes considered as anomalous behavior resulting in 5 different failure modes. Performance may be improved by adding sub-classes as more field data becomes available and failure mode information becomes evident.

Results from this study are encouraging as it demonstrates the value of conducting action design research using DSR. Experimental results will be used to enhance the real-world system by automatically identifying anomalies in the communication system and issuing work orders based on failure (or anomalous) events. Due to imperfections, there will still be human intervention to validate events, but this is expected and will be used to improve the system as more operational knowledge is obtained. Overall, the purpose of this research was achieved, namely to create a base from which a business intelligence system can be built for improvement of service availability.

VII. AUTHORS AND AFFILIATIONS

Johann Holm (PhD, PrEng) is an associate professor at the School for Electrical, Electronic and Computer Engineering of the North-West University in South Africa. He is a member of the IEEE and a member of INCOSE, and is actively involved in research and development, as well as consulting in operations and product development.

Liaan Moolman (BEng) is a postgraduate student in pursuit of the degree Master of Engineering at the North-West University in South Africa. He is involved as a researcher on the system presented in this paper.

Rossouw van der Merwe (PhD) is the managing director of Jericho Systems Pty Ltd, a product development company that specializes in IoT systems. He is involved as a development manager on the system presented in this paper.

VIII. ACKNOWLEDGEMENTS

The authors acknowledge Jericho Systems Pty Ltd for their financial support and for making an IoT platform available for this research.

IX. REFERENCES

REFERENCES

[1] B. Ramesh and A. Ramakrishna, "Unified business intelligence ecosystem: A project management approach to address business intelligence challenges," in *PICMET 2018 - Portland International Conference on Management of Engineering and Technology: Managing Technological Entrepreneurship: The Engine for Economic Growth, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 10 2018.

[2] E. Turban, R. Sharda, D. Delen, and D. King, "Introduction to business intelligence," *Business intelligence: a managerial approach*, pp. 3–18, 2011.

[3] T. Liyang, N. Zhiwei, W. Zhangjun, and W. Li, "A conceptual framework for business intelligence as a service (SaaS BI)," in *Proceedings - 4th International Conference on Intelligent Computation Technology and Automation, ICICTA 2011*, vol. 2, 2011, pp. 1025–1028.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. September, pp. 1–58, 2009.

[5] S. I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis, "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting," in *2016 IEEE International Energy Conference, ENERGYCON 2016*. Institute of Electrical and Electronics Engineers Inc., 7 2016.

[6] Y. Heryadi and H. L. H. S. Warnars, "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM," in *2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings*, vol. 2017-November. Institute of Electrical and Electronics Engineers Inc., 3 2018, pp. 84–89.

[7] Y. Lei, "Network Anomaly Traffic Detection Algorithm Based on SVM," in *Proceedings - 2017 International Conference on Robots and Intelligent System, ICRIS 2017*. Institute of Electrical and Electronics Engineers Inc., 11 2017, pp. 217–220.

[8] S. Ben-David and S. Shalev-Shwartz, "Understanding Machine Learning: From Theory to Algorithms," Tech. Rep., 2014. [Online]. Available: http://www.cs.huji.ac.il/ shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

[9] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," Tech. Rep.

[10] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, 3 2015.

[11] Y. Sasaki and R. Fellow, "The truth of the F-measure," Tech. Rep., 2007. [Online]. Available: https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf

[12] A. Hevner, "A Three Cycle View of Design Science Research U-CARE View project Modeling Customer Churn View project," Tech. Rep., 2014. [Online]. Available: https://www.researchgate.net/publication/254804390

[13] Sein, Henfridsson, Purao, Rossi, and Lindgren, "Action Design Research," *MIS Quarterly*, 2011.

[14] M. T. Mullarkey and A. R. Hevner, "Entering action design research," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9073. Springer Verlag, 2015, pp. 121–134.

[15] J. Han, M. Kamber, and J. Pei, "Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)," Tech. Rep., 2011.

IEEE

AFRICON - 2019

ACCRA, GHANA

Powering Africa's Sustainable Energy for All Agenda: The Role of ICT and Engineering.

region 8 IEEE

Certificate of Presentation

IEEE GHANA SECTION

This is to certify that the paper titled

Cloud-Based Business Intelligence for a Cellular IoT Network

and authored by

Johann Erich Wolfgang Holm, Liaan Moolman
and Gabriël Petrus Rossouw van der Merwe

was presented at the
"IEEE AFRICON-2019 International Conference" held at
GIMPA Executive Conference Centre, Accra, Ghana
on 25th - 27th September, 2019.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. FOUNDED NEW YORK 1884

Dr. Eric Kuada	Dr. George Eduful	Dr. Jeffrey Too Chuan Tan
Conference Chair	Section Chair	Session Chair