DM

Public Mobility Dashboard
A Digital Shadow Approach for EV Charging Monitoring at City-Level

MASTER DISSERTATION

Gustavo Manuel Nascimento Miranda
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

February | 2026

Public Mobility Dashboard
A Digital Shadow Approach for EV Charging Monitoring at City-Level

MASTER DISSERTATION

Gustavo Manuel Nascimento Miranda
MASTER IN INFORMATICS ENGINEERING

SUPERVISOR
Amâncio Lucas de Sousa Pereira

CO-SUPERVISOR
Filipe Magno de Gouveia Quintal

Faculdade de Ciências Exatas e da Engenharia

Master of Science Degree in Informatics Engineering

Public Mobility Dashboard: A Digital Shadow Approach for EV Charging Monitoring at City-Level

Gustavo Manuel Nascimento Miranda

Supervised by:

Dr. Amâncio Lucas de Sousa Pereira

Prof. Filipe Magno de Gouveia Quintal

Constituição do júri de provas públicas:

Dra. Karolina Baras, Presidente

Dr. Paulo Bala, Vogal

Dr. Lucas Pereira, Vogal

fevereiro de 2026

Resumo

A rápida expansão das frotas de veículos elétricos (VE) introduziu desafios significativos na gestão de infraestruturas urbanas, com particular incidência na estabilidade da rede elétrica e no acesso equitativo aos recursos públicos de carregamento. Esta tese apresenta a proposta, a implementação e a validação de uma plataforma **Digital Shadow (DS)** baseada em microsserviços, aplicada ao contexto operacional do Funchal, Ilha da Madeira. O sistema atua como uma camada centralizada de suporte à gestão, agregando fluxos de dados diversos provenientes da rede pública Mobi.e para fornecer visibilidade situacional em tempo real e análise detalhada do histórico de utilização.

Para além desta solução de monitorização, a tese aborda a limitação associada ao fenómeno de "Energy Blindness", caracterizado pela falta de telemetria de energia em tempo real nas infraestruturas públicas, por meio do desenvolvimento de um **Mecanismo Híbrido de Deteção de Anomalias**. Ao integrar o algoritmo **K-Means Clustering** para obter o perfil de utilização e o **Isolation Forest** para a deteção de valores atípicos, o sistema estabelece uma pipeline de calibração automática que identifica a "Ocupação Passiva" (estacionamento abusivo), inferindo anomalias exclusivamente com base em métricas temporais.

Os resultados experimentais demonstram que a arquitetura híbrida proposta supera os métodos convencionais, alcançando precisão superior à dos modelos isolados na identificação de comportamentos de carregamento irregulares. Estes resultados validam a eficácia do sistema para a supervisão municipal automatizada, filtrando eficazmente o ruído operacional e identificando infrações com grau de confiança assinalável. Adicionalmente, a verificação funcional do "Digital Shadow" confirma que as abstrações espaciais, estatísticas e temporais da plataforma reduzem significativamente a carga cognitiva dos operadores, transformando logs de API não processados em indicadores de suporte à decisão, por meio de mapas de calor geoespaciais e da reconstrução de sessões passadas.

Palavras-chave: Digital Twin, Digital Shadow, Veículos Elétricos, Deteção de Anomalias, Visualização de dados.

Abstract

The rapid expansion of electric vehicle (EV) fleets has introduced significant challenges for urban infrastructure management, particularly regarding grid stability and equitable access to public charging resources. This thesis presents the design, implementation, and validation of a microservice-based **Digital Shadow (DS)** platform applied to the operational context of Funchal, Madeira Island. The system serves as a centralised management support layer, ingesting heterogeneous data streams from the Mobi.e network to provide real-time situational awareness and historical usage analytics.

Beyond this monitoring solution, the thesis addresses the constraint of "Energy Blindness", characterised by the lack of real-time energy telemetry in public infrastructure, through the development of a **Hybrid Anomaly Detection Engine**. By integrating **K-Means Clustering** for behavioural profiling and **Isolation Forest** for outlier detection, the system achieves a self-calibrating pipeline that identifies "Idle Occupancy" (abusive parking), inferring anomalies exclusively from temporal metrics.

Experimental results demonstrate that the proposed hybrid architecture outperforms standalone methods, achieving superior precision in identifying irregular charging behaviours. These findings validate the system's reliability for automated municipal supervision, effectively minimising false positives while isolating high-confidence violations. Furthermore, functional verification of the Digital Shadow platform confirms that its spatial, statistical, and temporal abstractions successfully reduce cognitive load for system operators, converting heterogeneous data streams from raw API logs into meaningful decision-support insights through geospatial heatmaps and reconstructed session objects.

Keywords: Digital Twin, Digital Shadow, Electric Vehicles, Anomaly Detection, Data Visualisation.

Acknowledgments

I would like to express my sincere gratitude to all those who supported me throughout the course of this dissertation and the research project that provided the basis of it.

To Lucas Pereira, my academic supervisor, I extend my profound gratitude for his expert advice, technical insights, and continuous availability. Such support was indispensable, defining the direction of this thesis and ensuring the successful realisation of its objectives.

To the collaborative group I joined within the scope of the European Union's AHEAD project, and in particular the members of the Portuguese demo team, I am grateful for the support and cooperation provided during the practical development of this work. I also acknowledge the Universidade da Madeira and the Faculdade de Ciências Exatas e da Engenharia for the institutional support and facilities that enabled this research.

At last, to my family and friends, I owe the deepest thanks. For standing by me through the highs and lows of this chapter, and for offering the support and encouragement I often needed, I am deeply grateful. This thesis would not have been possible without you.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AHEAD AI-informed Holistic Electric Vehicles Integration Approaches for Distribution Grids

API Application Programming Interface

CBM Condition-Based Maintenance

CPU Central Processing Unit

CS Charging Station

DSO Distribution System Operator

DT Digital Twin

DTI Digital Twin Instance

DTP Digital Twin Prototype

DS Digital Shadow

EU European Union

EV Electric Vehicle

EVSE Electric Vehicle Supply Equipment

GPU Graphics Processing Unit

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

KPI Key Performance Indicator

LAN Local Area Network

ML Machine Learning

MQTT Message Queuing Telemetry Transport

NASA National Aeronautics and Space Administration

NFR Non-Functional Requirement

OCPP Open Charge Point Protocol

RUL Remaining Useful Life

SCADA Supervisory Control and Data Acquisition

SoC State of Charge

SoH State of Health

SPA Single Page Application

SQL Structured Query Language

SSOT Single Source of Truth

UI User Interface

1 Introduction

With the automotive industry increasingly investing in sustainable and low-emission solutions, and supported by government incentives worldwide, the global fleet of electric vehicles (EVs) has expanded rapidly in recent years. This growth has created an urgent need to extensively expand EV charging infrastructure to support mass adoption and ensure adequate service levels for users [1]. However, this rapid expansion presents a challenge in ensuring user convenience through accessible charger placement, while a spatially concentrated increase in EV charging demand raises new significant challenges for power systems, including local grid congestion, voltage control issues and the need for costly reinforcement in distribution networks, particularly in urban areas with high charger density [2].

At the present time, the transport sector is one of the largest contributors to greenhouse gas emissions and remains a critical obstacle to achieving climate neutrality in the European Union (EU). Road transport alone is responsible for around a quarter of the EU's greenhouse gas emissions, with passenger cars accounting for a dominant share, making its electrification a priority in the European Commission's strategy. To address this, the European Commission has defined ambitious decarbonisation goals for the vehicle fleet, including the reduction of emissions from transportation by 55% for new vehicles by 2030, compared to 2021 levels and achieve zero emissions by 2035.

As a result of this context, the number of light electric vehicles in the EU is expected to increase substantially over the coming decade. Projections indicate that the light EV fleet in the EU will grow from approximately 10.3 million vehicles to around 30 million by 2030. This growth will require a robust and well-planned charging infrastructure, capable not only of meeting the expected increase in energy and power demand but also of doing so without compromising grid stability, service quality, or user convenience [1].

1.1 Context

1.1.1 The AHEAD Project

To address the challenges of grid congestion, alongside the needs for intelligent infrastructure planning, the AI-informed Holistic Electric Vehicles Integration Approaches for Distribution Grids (AHEAD) project was established. At its core, this initiative seeks to build a simulation environ-

[1] `https://horizon-ahead.eu/`

ment capable of predicting the most convenient locations for EV charging stations (EVSEs) while simultaneously optimising the usage of power grid resources, ensuring that new infrastructure meets user needs without compromising network stability.

To validate possible solutions and approaches across diverse regulatory and geographical landscapes, the project deploys three complementary demonstrators. A Nordic Europe demo focuses on hybrid resources and workplace charging, a Southern Europe demo concentrates on heavy-duty vehicles and public transportation fleets, and a Western Europe demonstration targets the coordination of public and private charging infrastructures in urban environments. Given the distinct geographical and operational contexts of each region, these three demonstrators operate as completely independent workstreams. Consequently, the Nordic and Southern initiatives have no overlap or involvement with the activities conducted in the Western demonstrator.

1.1.2 The Western Demonstrator and Data Provenance

In particular, the Western Demonstrator includes specific activities in Funchal, Madeira Island, which serve as the practical case study for this research. These initiatives address the management of public charging infrastructure by reconciling the divergent goals of the Municipality and the Distribution System Operator (DSO), as the Municipality prioritises city-wide charger availability while the DSO safeguards the stability of the power grid.

Within the scope of the Funchal case study, it is essential to establish the functional boundaries of this thesis relative to the collaborative efforts of the team. An important step for this project was developed by Zakaria, another researcher involved in the AHEAD project, who developed the primary data acquisition layer. Specifically, the original data stream from the Mobi.e network operates through a non-conventional push-based API, requiring an initial handshake followed by continuous data pushes. Zakaria's contribution consisted of developing the infrastructure to intercept this live stream, persist the raw records into a structured database, and expose these records through a standardised, RESTful API.

Also, to effectively manage this public infrastructure, it is crucial to clarify the provenance, coverage, and limitations of the data sources used in this research. The project relies on two distinct streams. First, the Municipality of Funchal (CMF) provided historical Excel datasets for the chargers under their direct ownership, which include valuable session-level energy metrics. Second, the system consumes data from Mobi.e, the national network operator, both for real-time monitoring

and to access historical records for periods and chargers not covered by the CMF's files. It is important to explicitly state the limitations of this Mobi.e dependency early on: while it allows for broader network coverage and longitudinal analysis, the Mobi.e API strictly lacks energy telemetry (such as live amperage or energy consumption per session).

1.1.3 Municipal Management Needs and the Digital Shadow

Consequently, the work presented in this thesis builds directly upon that data foundation, focusing exclusively on the Municipality's perspective regarding city-level planning and optimising the management of the public charging services. Currently, the CMF manages this network through a highly reactive and manual process. The existing management relies entirely on raw operational data exported in Excel format and delivered on a monthly basis. Such a static methodology prevents any immediate response to operational inefficiencies or abusive behaviours.

To address these management limitations, a Digital Shadow (DS) is developed with the main objective of dynamically monitoring and tracking the operational status of the infrastructure. Simultaneously, the proposed solution addresses the concerns regarding the equitable utilisation of these public resources. In this regard, the proposed solution aims to identify inefficiencies caused by abusive behaviours, in particular the occupation of charging stations as parking slots.

1.2 Thesis Objectives

To address the necessity for informed decision-making and the current limited monitoring capabilities, the primary objective of this thesis is the design and implementation of a DS architecture aligned with the specific constraints of the Funchal municipality. This solution acts as the central aggregation point capable of ingesting and processing heterogeneous data streams, serving as the foundation for the following specific functional objectives:

- **Real-time Monitoring and Visualisation :** The development of a centralised dashboard built upon the DS's real-time data stream to provide immediate situational awareness regarding the operational status of the EV chargers, including their availability, occupation time and operational failure tracking, integrated through a geospatial visualisation layer.
- **Historical Data Analytics:** The implementation of data aggregation mechanisms designed to generate key performance indicators (Key Performance Indicator (KPI)s) regarding the

utilisation of the public charging network service, specifically through the quantification of occupancy rates, the identification of high-demand time windows and the profiling of daily utilisation patterns based on temporal and status data.

– **Identification of Abusive Behaviours:** The development of distinct data models and logical algorithms, with a particular focus on identifying "idle occupation" scenarios, characterised by the continued occupation of the EVSE after the charging process is complete.

1.3 Document Structure

This thesis is organised into eight chapters, guiding the reader from theoretical foundations to the practical implementation and validation of the system.

Chapter 2: Related Works establishes the theoretical framework, reviewing DT concepts, microservices architectures, and EV infrastructure challenges to justify the use of Unsupervised Machine Learning (ML) for anomaly detection.

Chapter 3: Proposed Solution presents the “Digital Shadow Ecosystem” conceptual design. It specifies the system’s Functional and Non-Functional Requirement (NFR) requirements and outlines the Distributed Microservices Architecture selected for the project.

The implementation is detailed across two chapters, separating analytical logic from software engineering:

Chapter 4: Data Profiling and Anomaly Detection Strategy details the analytical strategy for identifying “Idle Occupancy.” It describes the rule-based baseline and the data pipeline regarding the Hybrid Strategy, combining K-Means profiling with Isolation Forest detection.

Chapter 5: Implementation of the Digital Shadow Platform documents the platform’s technical materialisation. It details the engineering of the UI, API Gateway, and the backend services responsible for monitoring and visualisation.

Chapter 6: Evaluation Methodology defines the validation protocols. It establishes statistical metrics (Precision, Recall, F1-Score) for the detection engine and outlines scenarios to evaluate the platform’s data abstraction capabilities.

Chapter 7: Results and Discussion presents the empirical findings. It benchmarks the Hybrid Model against standalone algorithms and validates the DS's efficacy in transforming raw API data into actionable insights.

Finally, **Chapter 8: Conclusion and Future Work** summarises the main findings and limitations, such as "Energy Blindness," and proposes future research directions including the integration of Smart Meter telemetry.

2 Related Works

This chapter establishes the theoretical and technological framework for the thesis. The narrative begins by defining the DT paradigm and addressing its associated complexity through **Data Visualisation** strategies. Once the conceptual and interface layers are defined, the review moves to the **Microservices Architecture** required to support such a distributed system.

With the architectural foundation established, the discussion shifts to the specific requirements of the application domain, the **Electric Mobility**. Within this domain, specific operational constraints, in particular the lack of energy telemetry, are identified as the main limiting factor to efficient management. Finally, the chapter concludes by reviewing **Unsupervised Machine Learning** techniques, specifically K-Means and Isolation Forest, as the necessary analytical tools to overcome these data limitations and detect operational anomalies, despite the constraints.

2.1 Foundations of Digital Twins

2.1.1 Evolution of the Concept

The concept of the DT has evolved significantly since its introduction. In early 2012, one of the first technical papers on DTs was published by the National Aeronautics and Space Administration (NASA) [3]. In this reference document, DTs were described as a "multiphysics, multiscale, probabilistic simulation of a system that uses the best available physical models, sensor updates, fleet history to mirror the life of its corresponding twin" [3]. However, the origins of this technology can be traced to earlier work by Michael Grieves and John Vickers in 2003 on product lifecycle management, who initially conceptualised the idea of connecting physical products in real space with their virtual representations in virtual space [4], as illustrated in Figure 1.

Building upon its historical origins, modern literature clarifies that a DT is far more than a simple digital replica. According to a consolidated definition established from the academic research conducted by VanDerHorn and Mahadevan, it is "a virtual representation of a physical system, its associated environment and processes that is updated through the exchange of information between the physical and virtual systems" [5]. Complementing this structural view, Jafari emphasises the operational utility of the technology, stating its primary aim is "to provide better or the same information about the physical twin that can be achieved by having a physical entity, regardless of modelling, and simulation of physical system behaviour" [6]. In essence, this establishes the DT as

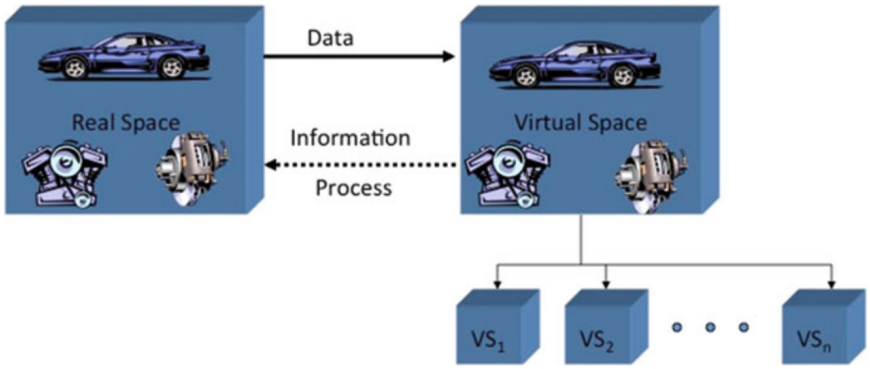


Figure. 1: The Digital concept established by Grieves and Vickers [4].

a powerful tool for enhancing analysis, monitoring, and decision-making processes for the physical system.

2.1.2 Historical Applications

The transition moving from academic theory to practical implementation is best observed through the technology's historical adoption curve. While the core principle of a virtual mirror remains consistent, the operational scope has progressively expanded from isolated high-value assets to complex, interconnected systems.

In the aerospace sector, where the concept originated, the DT became the enabling technology for Condition-Based Maintenance (CBM). Unlike traditional schedules, which replace parts after a fixed number of hours regardless of their actual condition, CBM dictates that maintenance should occur only when objective indicators indicate decreased performance. NASA utilised the DT to bridge this gap by feeding sensor data into a physics model, which enabled them to simulate internal structural stresses that were impossible to measure physically. This allowed engineers to "see" the remaining fatigue life of a specific physical asset, preventing failures without performing unnecessary repairs [3].

As IoT ecosystems evolved, the energy sector also adapted this logic to manage essential energy storage infrastructure, specifically Lithium-Ion Battery Systems. Expanding upon monitoring, researchers implemented Cloud-based DTs to predict the internal degradation of battery cells. In this domain, the technology's focus shifted from fault detection to predictive lifespan maximisation. By running electrochemical models in the cloud, which are computationally too intensive for a vehicle's onboard chip, operators could estimate the "State of Health (SoH)" and "Remaining Useful

Life (RUL)" of the battery with high precision. This allowed fleet managers to predict exactly when a battery would fail months in advance, optimising replacement schedules and preventing unexpected downtime [7].

Most relevant to this thesis, however, is the recent expansion into the Smart City domain, which represents a shift towards managing distributed "systems of systems". Projects like the Helsinki 3D+ initiative established the precedent of utilising standardised models to create a representation of the entire city. Unlike previous industrial iterations focused on mechanical failure, this urban twin was utilised to simulate city-wide energy phenomena, such as calculating the solar potential of thousands of rooftops to guide municipal energy planning [8]. This progression, from monitoring a single physical part to simulating distributed urban infrastructure, serves as an operational basis for the EV charging monitor proposed in this work.

2.1.3 Digital Twin Model

Accompanying the widespread adoption of these systems in recent years is the lack of a standardised definition. The literature contains several characterisations, each tailored to specific use cases, making it difficult to establish a universal standard [5]. To address this challenge, VanDerHorn and Mahadevan proposed a consolidated model establishing three fundamental components that serve as a framework to guide definition and implementation:

The foundation of the framework is the **Physical Reality**, which represents the physical object and its context. Crucially, this concept encompasses the entire system, including both known and unknown components. At its core lies the Physical System, a set of interacting entities that form a unified whole, situated in a specific Physical Environment. Within this context, Physical Processes express the mechanisms that determine how the system changes state. The measurement of these state changes serves as the fundamental input for virtual simulations and forecasts.

Mirroring this physical entity in the digital space is the **Virtual Representation**, constructed as an idealised form of the physical reality [5]. This representation relies on two primary model types: *Data Models*, which serve as structural representations containing all variables at a chosen level of abstraction, and *Behaviour Models*, which define how these variables connect and interact. Mirroring the physical world, the Virtual System integrates these models to simulate physical counterparts, while Virtual Processes describe behaviour based on input-output relationships. This

simulation capability enables insights into future states, facilitating diagnostics, design verification, and scenario analysis [5].

Finally, enabling the dynamic synchronisation between these two worlds are the **Interconnection Mechanisms**, which facilitate bidirectional information exchange between the physical and virtual domains [5], represented by Figure 2. The *Physical-to-Virtual* Connection ensures the continuous alignment of the digital model with real-world conditions by executing a process that begins with data collection from the physical environment, proceeds to the interpretation of these metrics, and culminates in updating the system state. Completing the cycle, the *Virtual-to-Physical Connection* closes the DT loop, enabling insights, forecasts, and decisions generated in the virtual space to be implemented in the physical system to optimise performance.

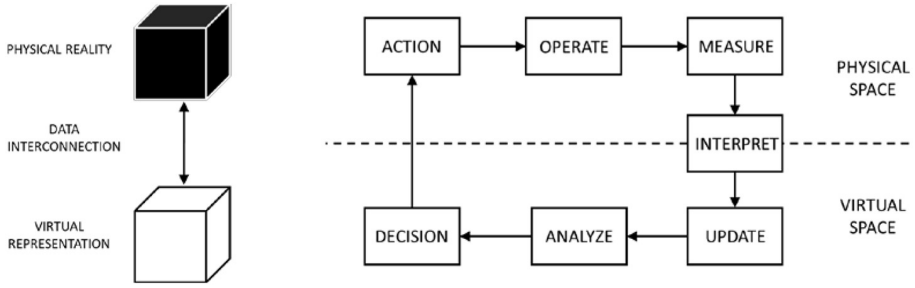


Figure. 2: The DT Reference Model illustrating the bidirectional data flow between the Physical and Virtual entities [5].

2.1.4 Types of Digital Twins

While the fundamental components remain consistent, the specific nature of a DT evolves throughout a product's existence. To better understand the operational scope of these systems, researchers distinguish between classifications based on the asset's lifecycle stage.

The **Digital Twin Prototype (DTP)** serves as the conceptual specification for a physical product. It contains all the data necessary to describe and produce the physical version, essentially acting as a comprehensive set of defining characteristics. A DTP does not refer to a specific physical object, but rather to the abstract design of the product class itself, representing the idealised properties that exist prior to the creation of any individual physical instance [4].

In contrast, the **Digital Twin Instance (DTI)** corresponds to a specific, individual physical product once it has been manufactured. The DTI remains linked to its physical counterpart throughout

its entire lifecycle, accumulating unique operational data, history, and usage patterns [4]. For example, while the DTP defines how a charger model behaves in theory, the DTI captures how a specific charger at a specific location performs in reality. Consequently, given the focus of this thesis on monitoring existing deployed assets (EV chargers) in Funchal municipality, the **DTI** is the primary reference point.

2.1.5 The Hierarchy of Digital Representations: Models, Shadows, and Twins

A critical aspect of the theoretical framework is the nature of the data flow between the physical and virtual counterparts. While the term "Digital Twin" is frequently used as a broad designation in industrial applications, the literature establishes a distinct hierarchy based on the level of data integration and synchronisation [9].

At the foundational level lies the **Digital Model**. According to the literature, this representation provides only a "snapshot" of an object at a specific moment [10]. Unlike the dynamic nature of its counterparts, a Digital Model captures the system state at a fixed temporal instance, lacking the continuous temporal extension that characterises both Shadows and Twins.

Moving up the hierarchy, a **DS** is characterised by an automated one-way data flow from the physical object to the digital object. A change in the state of the physical system leads to an immediate change in the digital representation, but the digital system cannot automatically influence the physical object [9]. This architecture is primarily designed for monitoring, visualisation, and analysis.

Finally, a fully implemented **Digital Twin** requires bi-directional communication and data flow (Figure 3), enabling active interaction between physical and virtual domains through online data interconnection [5, 9]. Through this design, the digital object not only receives data but can also affect the physical object, effectively closing the control loop. As noted in the literature, a system must possess this active control capability to meet the strict definition of a Twin [9].

Distinguishing these concepts is fundamental to defining the technological limitations and capabilities of this work. While a full DT entails active control over the physical asset, the charging stations in Funchal constitute public infrastructure. Consequently, direct actuation or remote intervention by the municipality is restricted due to operational constraints. Since the system is technically

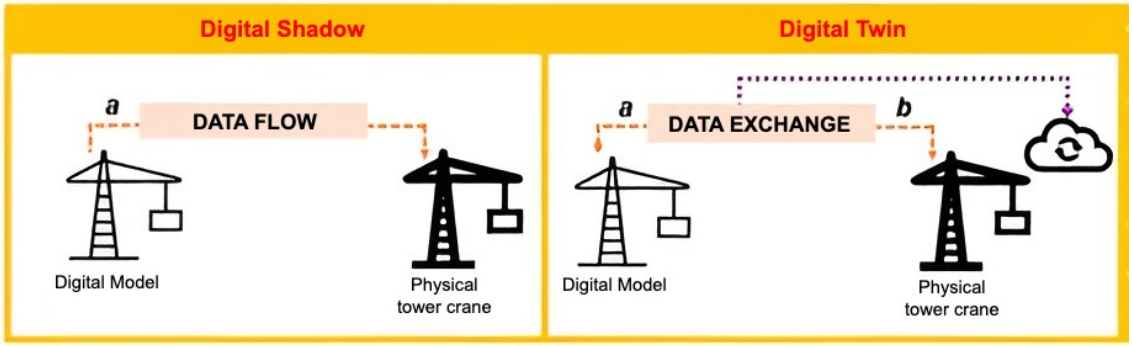


Figure. 3: Classification of digital representations based on the level of data integration: DS (unidirectional), and Digital Twin (bidirectional) [9].

limited to unidirectional data acquisition, without the ability to physically alter charging sessions, the architecture proposed in this thesis aligns with the **DS** concept.

2.1.6 Challenges and Limitations

Despite their projected value, the large-scale implementation of DTs and Shadows faces several technical and organisational challenges that must be addressed to fully exploit their capabilities.

A fundamental constraint is the **reliance on high-quality input**, as missing or incomplete data poses a significant risk. Gaps in sensor data, transmission failures, or fragmented historical records can significantly impact the performance and reliability of the entire system, potentially leading to inaccurate outputs [9]. Beyond the availability and reliability on precise data, system integration is significantly compromised by **lack of standardization**, as highlighted in section 2.1.3, the absence of a universal framework for defining and executing these models make more difficult the integration of systems from different technology providers, as each solution often adheres to proprietary rather than standardized architectural definitions [5].

Beyond these technical constraints, organisational barriers could similarly impede an effective deployment. In the context of smart cities, data is frequently fragmented across different entities, leading to **organisational siloing**. Limited collaboration among stakeholders, such as utility providers, municipalities, and private operators, creates isolated information repositories that prevent a comprehensive understanding of the system [5] [4]. Finally, the **complexity for users** remains a critical barrier to adoption. Transforming high-dimensional raw data into accessible information is difficult, and without effective abstraction, the simulations produced by a DT can be

challenging for non-domain experts to interpret [5], consequently reducing the practical usability of the system.

2.2 Data Visualisation for Digital Twins

Due to the significant complexity of DTs and DS, as previously outlined in Section 2.1.6, visualisation tools emerge as a solution to increase their acceptance and demonstrate their value to non-domain experts [5]. As these systems generate vast amounts of data, it is essential to have an interface that enables users to interact with [11] and analyse data effectively [12], transforming complex information into understandable content. In this context, insightful data visualisation becomes fundamental for enhancing the user experience and supporting decision-making processes [13].

2.2.1 Data Monitoring and Dashboards

The primary mechanism for supporting these decisions is effective data monitoring. By combining charts, tables, maps, and interactive timelines, digital twins provide a consolidated view of system performance (Figure 4) [12,13]. These tools are essential for tracking real-time data and identifying patterns and anomalies that might otherwise go unnoticed, enabling users to respond quickly to critical events [14]. Interactivity is a crucial aspect, allowing users to explore data at different levels of detail and customise visualisations according to their specific needs [15]. Dashboards, in particular, facilitate comprehensive analysis by allowing data filtering across multiple dimensions [12,16]. In the context of this thesis, relevant filtering criteria would include sensor type, location, and time period.

(a)

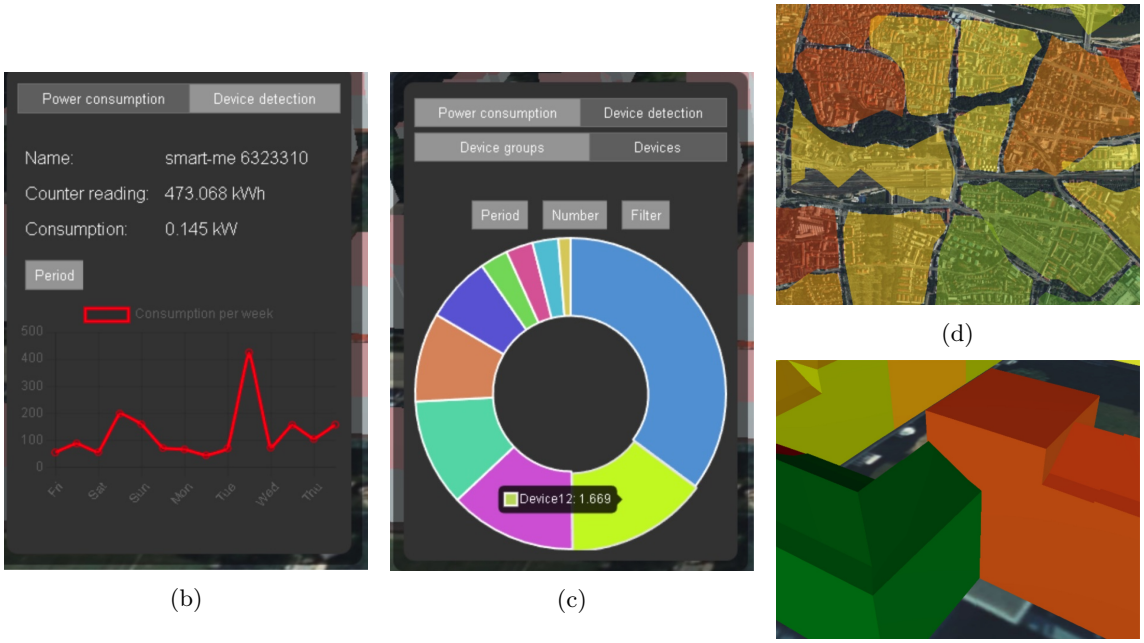


Figure. 4: Component breakdown of the DT dashboard, featuring various modules for real-time monitoring and data analysis [16] (Continued).

2.2.2 Geospatial Contextualisation

In synergy with these filtering capabilities, geospatial tools extend the monitoring capability by projecting data into a relevant context. Contextual maps, as the one in Figure 5 [16], for example, help visualise the distribution of variables, such as occupancy across different geographic areas, enabling the identification of spatial patterns and effectively transforming raw metrics into comprehensible and meaningful insights [12, 17]. As demonstrated in urban studies, interactive maps allow users to explore specific areas in greater detail [17], an essential feature for the geographical scope of this study.

2.2.3 Complexity Reduction

Complementing these analytical and spatial dimensions, visualisation serves as a powerful tool for Complexity Reduction, simplifying data interpretation [5]. Innovative resources, such as an immersive graphical interface, enhance the user experience by visually and intuitively presenting physical systems. For instance, instead of numerical tables, a visual representation enables users to quickly understand the system's status, identify issues, and recognise relationships within the data [12]. Furthermore, implementing different levels of detail [16], as exemplified in Figure 6 is crucial to prevent users from feeling overwhelmed by the vast quantity of data. This approach

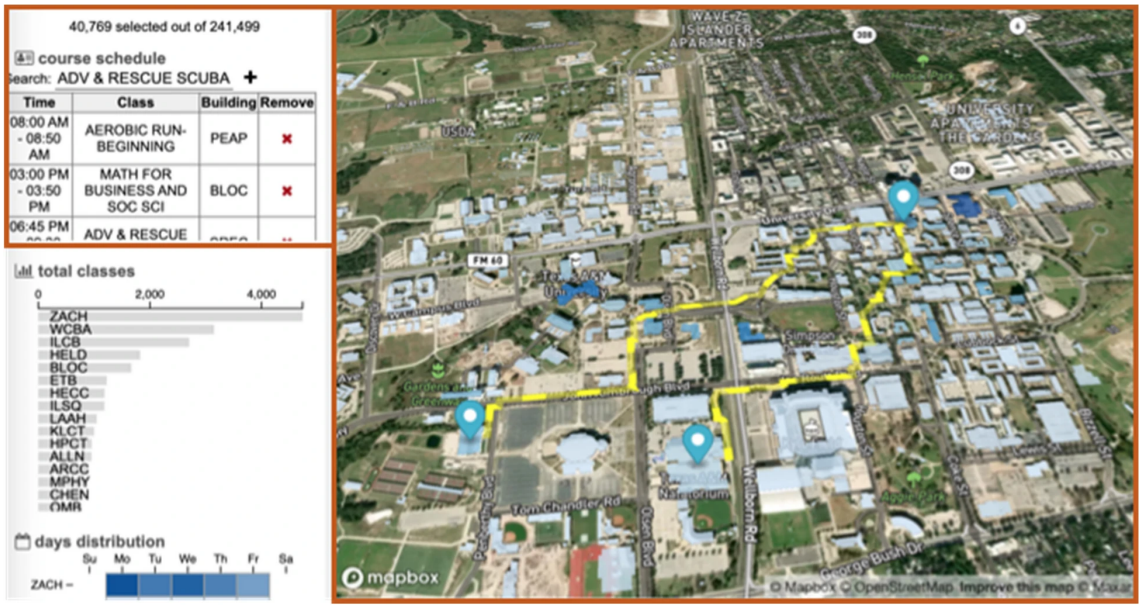


Figure. 5: Integration of DT assets into a geographic map for better context [12].

enables the system to accommodate the specific needs of diverse user profiles, whether they require a high-level strategic overview or precise technical monitoring. The visualisation adjusts to the specific context, ensuring the system remains accessible and useful across the entire organisational spectrum.

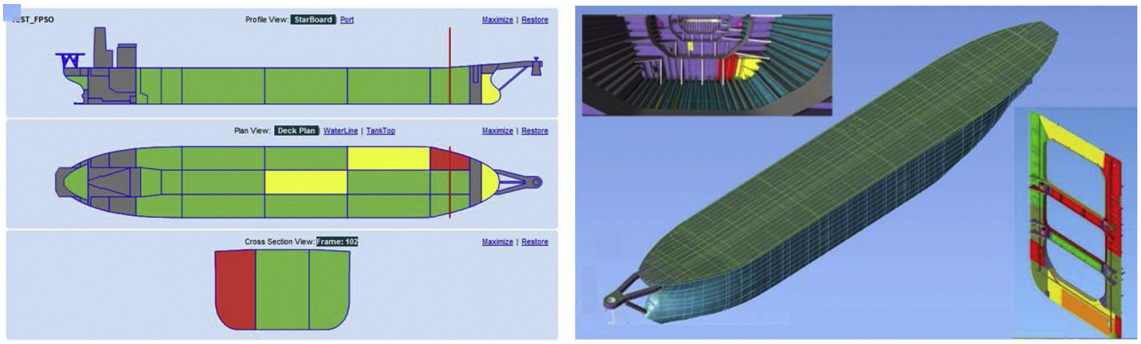


Figure. 6: Two representations of the same digital object showing different levels of complexity [5].

However, the effectiveness of these visualisations relies entirely on the availability and freshness of the supporting data. This requirement for a continuous stream of data creates specific scalability and maintainability challenges that lead directly to the architectural choices discussed in the following section.

2.3 Microservice Architectures for IoT and Digital Twins

While the previous sections established the theoretical components of DTs and DSs, and the visualisation strategies necessary for user interaction, the operational capability of any such system is fundamentally determined by its supporting architecture. Architecture refers to the structural design that defines how components interact, scale, and maintain consistency.

In the context of DTs and DS, architecture is not merely a procedural detail but a crucial determinant of operational success. Unlike traditional static applications, DTs must ingest heterogeneous data streams, process high-frequency updates, and maintain synchronisation between physical and virtual states in near real-time [5]. Consequently, the transition from isolated embedded systems to interconnected smart city platforms requires robust architectural designs capable of handling these specific demands [6]. This section explores the technological infrastructure and design strategies required to implement such systems effectively.

2.3.1 Technological Infrastructure and Protocols

A robust DT depends on a sophisticated technological infrastructure that supports accurate data collection, real-time communication, and seamless integration of information. This infrastructure ensures that the DT can effectively simulate, predict and optimise the performance of its physical counterpart, providing valuable insights for decision-making and system improvement [18]. However, unlike a full DT that actuates changes based on these insights, a DS operates exclusively as an observer. Consequently, although the architectural capacity for optimisation exists, the system prioritises accurate data ingestion and visualisation, providing a precise digital mirror without issuing control commands to the physical assets.

At the physical level of this infrastructure, sensor networks and communication protocols enable efficient data transmission. Industrial protocols like Modbus are commonly used for wired connections, such as RS-485, to interface directly with hardware. Local Area Network (LAN) transmit data within a local area using common TCP/IP protocols, ensuring local connectivity. On a larger scale, the internet enables global communication, with protocols such as Message Queuing Telemetry Transport (MQTT) and Hypertext Transfer Protocol (HTTP) managing data exchange between systems distributed across different locations. This hierarchical approach ensures that

data flows seamlessly between the physical and digital systems, enabling real-time updates and analysis [18].

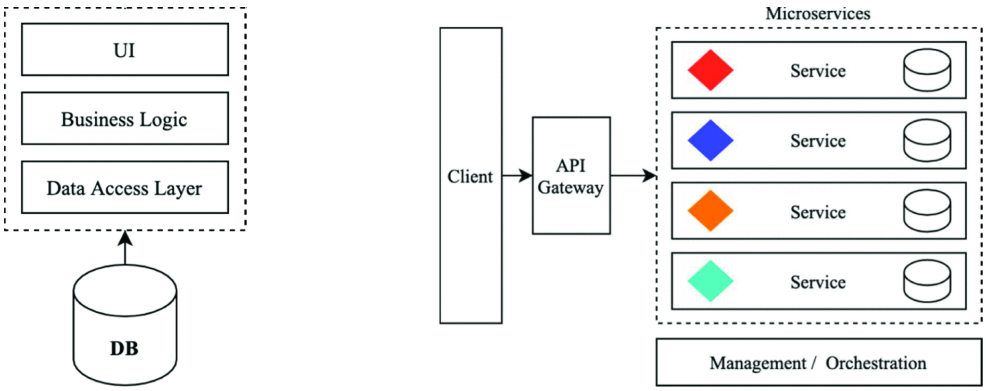
Building upon this foundation, the proposed platform resides at the application layer, abstracting this direct hardware communication. Instead, it relies primarily on the HTTP and RESTful APIs as the primary interfaces for data consumption. These protocols allow the application to interact with the system's backend services, such as querying historical data, managing user sessions, and retrieving real-time status, through a standardised web interface [19], detaching the user experience from the complexity of the low-level hardware and communication protocols.

2.3.2 Monolithic vs. Microservice Architectures

Historically, IoT platforms were often developed using Monolithic Architectures (Figure 7a), where the user interface, business logic, and data access layers are tightly integrated into a single deployable unit. While this approach offers simplicity in the early stages of development, the literature consistently highlights its limitations for complex, evolving systems. As noted by Dragoni's research, monoliths suffer from tight coupling, where a modification in one module can trigger unanticipated side effects in others, leading to a phenomenon often described as "dependency hell" [20]. Furthermore, monoliths impose a restrictive scaling strategy where individual components cannot be isolated [21]. Consequently, if a single function experiences high load, the entire application unit must be duplicated to handle it [20], forcing the allocation of unnecessary resources to parts of the system that are not being used.

To address these constraints, modern DT implementations increasingly adopt Microservice Architectures (Figure 7b). In this design methodology, an application is structured as a collection of modular, independently deployable services, each organised around a specific business capability [22]. This architectural style aligns naturally with the distributed nature of Smart Cities and DTs, specifically by addressing the need for independent scaling of specific components. For example, the simulation engine may need significant computational resources during a prediction cycle, while the user dashboard remains in a low-demand state. Microservices allow these components to scale independently, unlike a monolithic approach, where the entire system must scale together. Beyond scalability, research indicates that microservices significantly promote fault isolation, such that even if a specific analytic service fails (e.g., a heatmap), the core monitoring services remain operational, ensuring high system availability [23].

To support this modularity, the implementation of microservices relies on Containerization technologies, such as Docker. Containers provide a lightweight, portable execution environment that packages a service with all its dependencies. This ensures consistency across development and production environments and enables a multi-language development strategy, allowing developers to select the most appropriate language or database for each specific service without strictly locking the entire system into a single technology stack [24].



(a) Monolithic Architecture. (b) Microservices Architecture.

Figure. 7: Comparison between monolithic and microservices architectures [21].

2.3.2.1 Adoption in Digital Twin Architectures

This structural transition is well-documented in recent DT literature. For instance, a study by Nicolas et al. proposed a microservice-based architecture specifically for Smart Mobility, demonstrating that decoupling simulation engines from data ingestion services enables real-time traffic simulations without interrupting the data ingestion flow [25]. Additionally, the study identifies this approach as a fundamental prerequisite for system flexibility and scalability. By overcoming the rigid interdependencies inherent to monolithic designs, the proposed architecture enables independent scaling of heterogeneous services. This modularity ensures that the unified city-wide model remains adaptable to evolving urban dynamics, promoting a resilient ecosystem that can expand seamlessly. These precedents confirm that, for a system requiring scalability and continuous expansion, such as a public charging monitor, a distributed microservice architecture is not just a design choice but an operational necessity.

2.3.3 API Gateway Patterns and Data Orchestration

While this modularity, recognised in a microservice architecture, significantly enhances backend flexibility, the resulting decomposition of logic into numerous small services creates a structural complexity regarding service accessibility. Direct client-to-service communication, though valid in simpler contexts, represents a counterproductive approach within microservice architectures, as it increases latency, exposes internal endpoints, and compromises security [26].

To mitigate this, the API Gateway layer has emerged as a standard architectural component. Acting as a single entry point for all client requests, the Gateway handles System-wide operational tasks [26]. More importantly, it functions as an orchestrator, routing external requests to the appropriate microservices (Figure 8b). This abstraction layer isolates the client interface from the microservices' logic, allowing the internal architecture to change and evolve, by splitting, merging or managing service versions (Figure 8a) without impacting the client application.

(a) API: Service version management. (b) Service request routing.

Figure. 8: API Gateway [26].

2.4 EV Charging Management and Infrastructure

Having established the *architectural and technological foundations* (the Microservice-based Digital Shadow and its visualisation layer), it is necessary to define the **operational territory** where it will be applied. This thesis focuses on the EV charging infrastructure, a domain where the capabilities of a Digital Shadow, specifically remote monitoring and data abstraction, are critically needed.

As modern urban environments transition towards sustainable and interconnected smart cities, decarbonization, especially in the transportation sector, has become a primary objective. In this context, electric mobility has emerged as a key solution to reduce the environmental impact caused

by fossil fuels [1]. Driven by recent technological advances, particularly in lithium-ion battery technology, the sector has witnessed significant growth, with modern vehicles offering greater range and performance compared to earlier generations [27]. Alongside these improvements, this transition has been further supported by government incentives [28], including tax reductions, purchase subsidies, and funding for charging infrastructure [1, 28].

However, despite this progress, electric mobility faces critical operational challenges that limit widespread acceptance. A major obstacle is the insufficient charging infrastructure, particularly in densely populated urban areas and across long-distance routes [29]. This issue is aggravated by the limited availability of fast chargers, which could significantly reduce wait times but are not yet widely available, and by battery limitations, which, despite progress, still do not fully match the convenience of internal combustion vehicles [27].

Regardless of the speed of adoption, the integration of these vehicles introduces systemic risk, as the simultaneous, uncoordinated charging of a large number of vehicles during peak hours can overload the electrical grid, compromising stability and energy distribution capacity [30]. To mitigate these risks, management systems are essential for providing visibility into resource distribution and ensuring the efficient utilisation of limited infrastructure resources.

2.4.1 Role of Digital Twins in EV Mobility

DTs offer a promising technological solution to address challenges in charging infrastructure and grid stability. Their ability to analyse historical data enables continuous optimisation of EV charging processes. Complementing this analytical capability, the technology enables the virtual representation of real-world scenarios. This digital environment saves time and resources by facilitating virtual diagnostics, which is far more efficient, resource-saving, and less time-consuming than conducting field interventions [31].

Crucially, DTs provide continuous, real-time monitoring of systems, ensuring operators have secure, convenient access to information used in decision-making [27]. This improved visibility enables rapid responses to system changes or issues, enabling maintenance and management of the charging ecosystem.

2.4.2 The Open Charge Point Protocol (OCPP)

To enable the real-time monitoring described above, a standardised communication framework is required between the Electric Vehicle Supply Equipment (EVSE) and the central management system. The Open Charge Point Protocol (OCPP) has emerged as the industry reference standard for this purpose, enabling interoperability between hardware from different technology suppliers and monitoring solution platforms [32].

OCPP is a bidirectional protocol that governs all aspects of the charging session. For the purpose of DT modelling, three specific communication messages are of particular relevance:

- **StatusNotification:** Sent by the charger to report its current state (e.g., *Available*, *Occupied*, *Faulted*). This allows the DT to track the network's real-time availability.
- **StartTransaction / StopTransaction:** These messages mark the beginning and end of a charging session, providing the timestamps and user identification data necessary for session profiling.
- **MeterValues:** Transmitted during active charging sessions to facilitate energy monitoring, these values typically include current (A), voltage (V), power (W), and State of Charge (SoC).

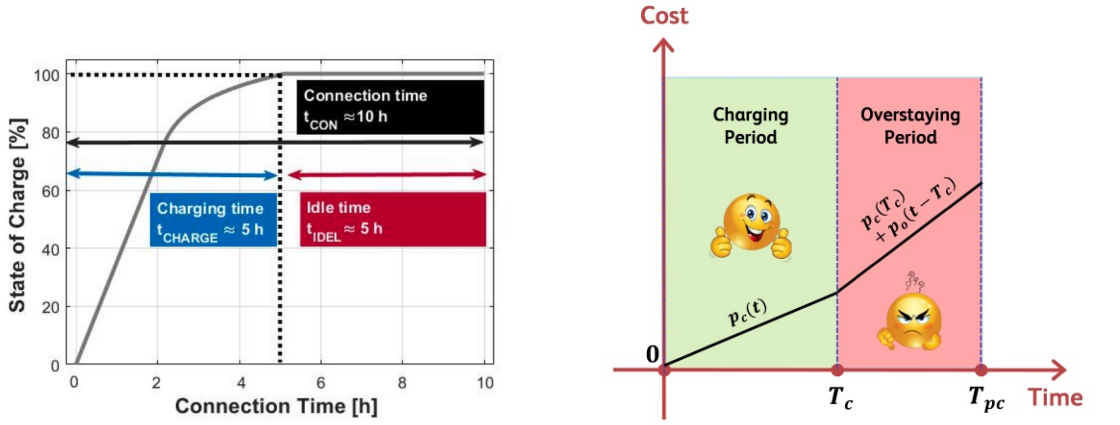
However, it is important to emphasise that, within the infrastructure analysed in this work, access to specific energy metrics is restricted by data constraints associated with the operational use of public charging networks. Consequently, the DT can construct a virtual representation of the physical charging process by continuously ingesting the content of those messages, allowing it to detect deviations from expected behaviour.

2.4.3 Operational Inefficiencies and Idle Occupancy

While protocols like OCPP ensure data connectivity, they do not necessarily solve operational behaviours that compromise service availability. A significant constraint in public charging networks is the phenomenon of "Idle Occupancy", which occurs when a vehicle remains connected to the Charging Station (CS) long after the battery has reached its full capacity (Figure 9a) [33], or after the active charging session has been concluded [34].

This behaviour artificially reduces the availability of the charging network, as the connector remains physically occupied (status *Occupied*) even though no energy is being transferred, potentially

triggering mitigation strategies (Figure 9b) [35]. Given the restricted access to energy metrics highlighted previously, traditional detection methods based on power drops are not feasible in this context. Consequently, identifying these overstaying events requires analysing session duration and user patterns rather than electrical throughput. As a result, there is a need for alternative approaches, such as anomaly detection, to detect these inefficiencies, offering a viable alternative for flagging overstaying events in the absence of real-time energy measurements.



(a) Distinguishing effective charging duration from idle connection time [33].

(b) Applying penalty pricing during the overstaying period [35]

Figure. 9: Characterisation of Idle Occupancy.

2.5 Data Profiling and Anomaly Detection

As established in Section 2.4.3, detecting operational inefficiencies, such as Idle Occupancy, cannot rely on energy metrics due to the restrictions imposed by the proposed architecture's integration with public infrastructure. Given the unavailability of real-time charging demand data to indicate the completion of a charge, the identification of abusive behaviour must shift from metric-based monitoring to behavioural analysis.

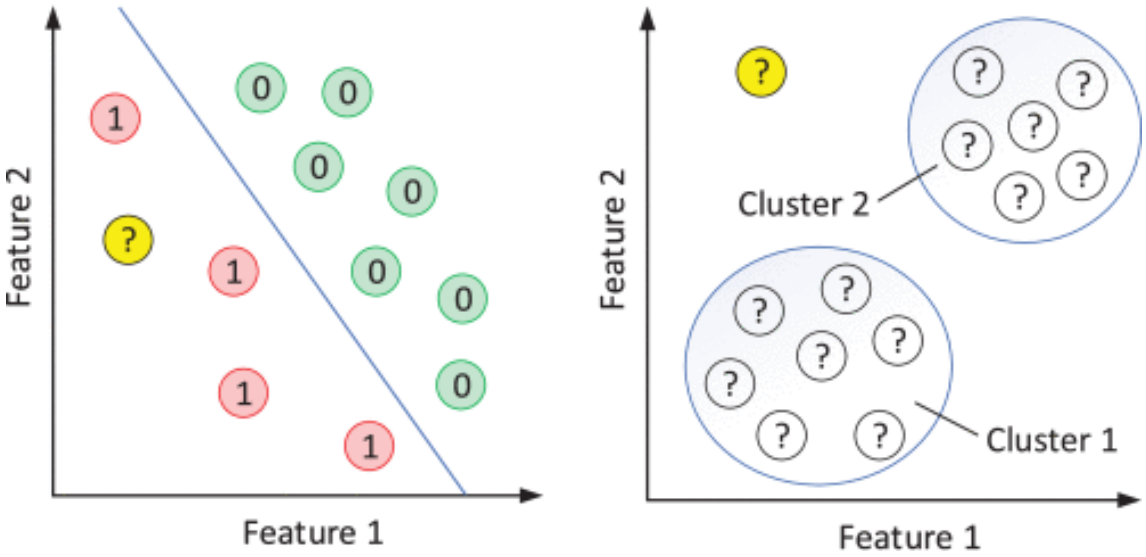
To address this challenge, this section explores the relevant analytical techniques for analysing the session parameters available through OCPP messages, specifically session duration. By utilising these available metrics, despite the lack of energy information, it becomes possible to establish "standard usage profiles" that serve as a baseline for identifying irregularities.

2.5.1 Unsupervised Machine Learning

A significant challenge in implementing this behavioural analysis is the nature of the available data. Traditional detection systems often rely on Supervised Learning (Figure 10a), which requires balanced datasets where anomalies are explicitly labelled. However, in real-world charging infrastructure, historical logs typically lack this "ground truth" [36, 37], as there is no pre-existing flag indicating which sessions were "abusive" or "normal".

Compounding this issue is the lack of a clear definition of abuse, since, without energy metrics to verify the real state of charge, there is no clear distinction between a legitimate long-duration session and an abusive idling event.

Consequently, literature in the field prioritises the **Unsupervised Learning** paradigm (Figure 10b) for these scenarios. Unlike supervised methods that search for labelled instances of abuse, unsupervised algorithms aim to discover the patterns and probability distribution of the data itself [36]. This approach assumes that the majority of recorded events reflect "normal" behaviour. By processing unlabelled historical data to establish a statistical baseline of normality, these methods can effectively identify data points that deviate significantly from the standard behaviour, allowing for the detection of previously unnoticed anomalies.



(a) Supervised Learning. (b) Unsupervised Learning.

Figure. 10: Learning Scenarios [36].

2.5.2 Clustering Algorithms: K-Means

To conduct this search for typical activity, the immediate task is to group similar sessions to identify standard usage patterns. Clustering is the fundamental technique used for this purpose. Among these, **K-Means** is one of the most commonly adopted segmentation methods due to its computational efficiency and scalability [38].

Mathematically, K-Means aims to decompose n observations into k clusters as illustrated in Figure 11, where each observation belongs to the cluster with the closest centre (centroid). The algorithm works by repeatedly adjusting the centre of each group to make the clusters as compact as possible. This ensures that all the sessions inside a single cluster are very similar to one another.

In energy analytics, K-Means is cited in the literature for **Load Profiling**, applied to categorise charging sessions into standard profiles (e.g., distinguishing between short-term daytime charging and long-duration overnight charging) [38]. By defining these standard clusters, the algorithm provides a reference point for separating behaviours.

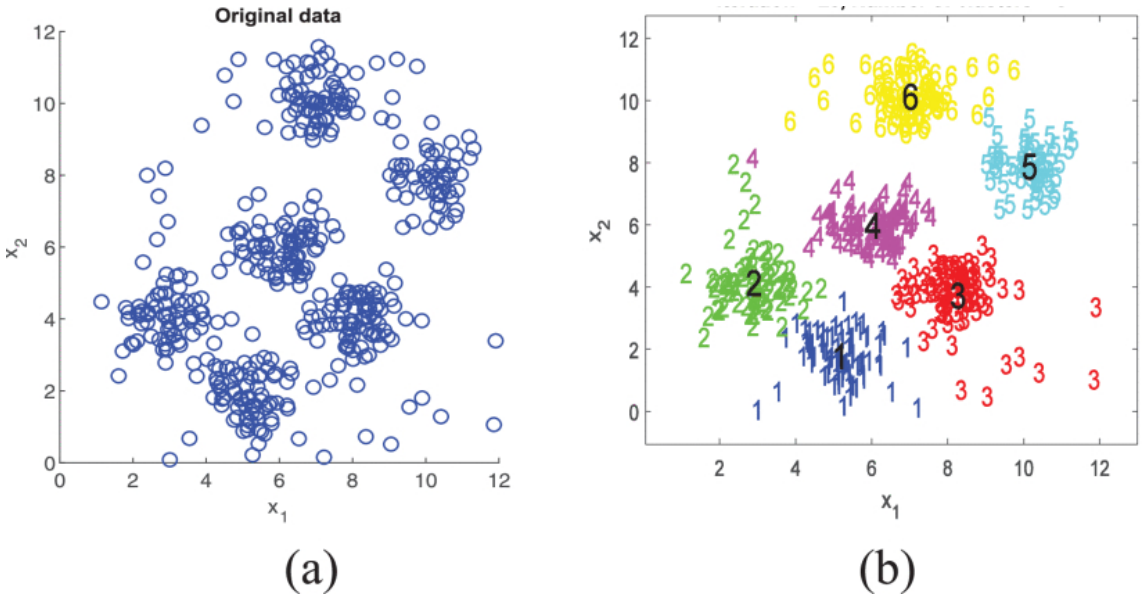


Figure. 11: K-Means algorithm Visual Representation [39].

In the specific domain of EV infrastructure, K-Means has emerged as a predominant technique for user segmentation. A study by Helmus et al. analysed over 4.9 million charging transactions, employing K-Means to successfully categorise users into distinct behavioural groups, classifying sessions by duration (short versus medium) and temporal distribution (daytime, overnight, or

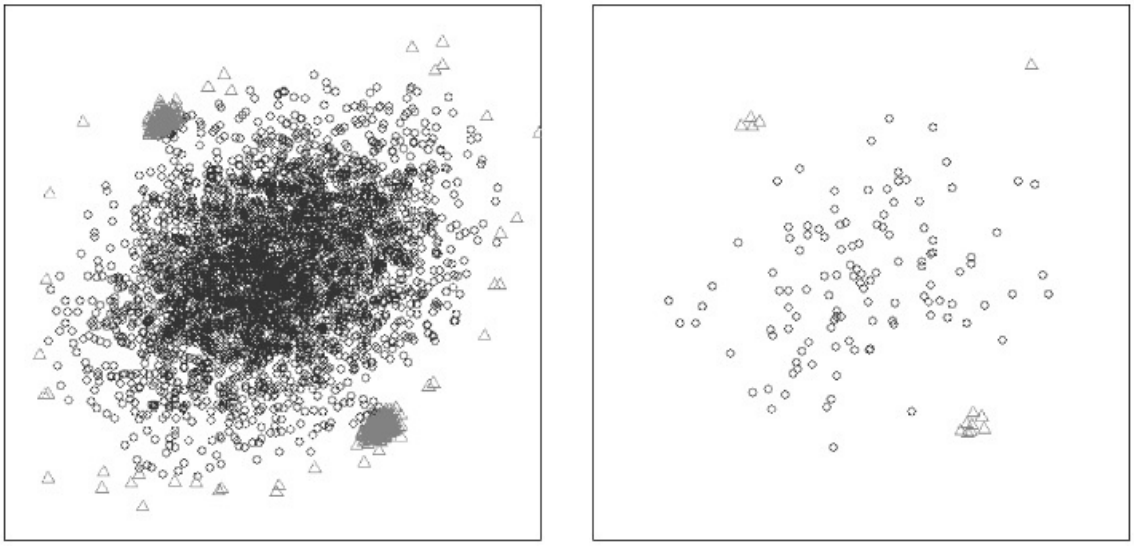
mixed) [40]. This work validates the premise for this thesis's approach, utilising K-Means not solely for a preliminary investigation, but to formally characterise the "Normal" usage profiles, providing a reference for outlier detection.

2.5.3 Outlier Detection: Isolation Forest

While clustering methods like K-Means are effective for grouping data, they are not always optimised for explicitly detecting anomalies, particularly in high-dimensional datasets. To address this, **Isolation Forest** was introduced as a method specifically designed for anomaly detection rather than profiling [41].

Unlike distance-based or density-based measures, Isolation Forest exploits the concept of isolation. The theoretical premise is that anomalies are few and distinct, making them more susceptible to isolation [42]. The algorithm constructs a series of random decision trees. In these trees, "normal" instances typically require a high number of splits to be isolated (resulting in a longer path length from the root). In contrast, anomalies differ significantly from the majority and are isolated very quickly (resulting in a shorter path length).

This approach is particularly valuable in IoT contexts as it is agnostic to the form of the data distribution and computationally efficient, making it suitable for identifying "irregular" sessions that do not perfectly align with any learned cluster, as illustrated in Figure 12.



(a) Complete Sample (4096 samples). (b) Sub-Sample (128 samples).

Figure. 12: Isolation Forest Visual Representation [42]

The need for distribution-independent methods for monitoring datasets has been comprehensively demonstrated in recent literature. In a study regarding Supervisory Control and Data Acquisition (SCADA) systems, Lin et al. integrated Isolation Forest to clean data for wind power prediction. The authors demonstrated that traditional statistical methods failed to accurately detect outliers because the operational data did not follow a Normal Distribution, a characteristic shared by EV charging profiles (which often exhibit hard cut-offs and non-uniform value concentrations). By utilising Isolation Forest, which relies on partitioning rather than statistical density, the authors achieved an effective anomaly filtration, reducing prediction errors [43]. This finding substantiates the choice of Isolation Forest for this thesis, confirming that it is uniquely suited for the non-linear, high-dimensional datasets typical of modern energy infrastructure.

2.5.4 Deterministic and Rule-Based Models

However, relying solely on these probabilistic outcomes introduces ambiguity. Distinct from the probabilistic nature of the ML models discussed above, Rule-Based Systems operate on deterministic logic. While ML algorithms deduce patterns from data distributions, rule-based approaches rely on explicit domain knowledge to define objective criteria for valid operation [44].

In the context of EV charging, these systems utilise established technical limits as reference points [44]. This research implements these rule-based methods as a comparative baseline for unsupervised learning models. By analysing discrepancies between statistical anomalies detected by K-Means or Isolation Forest and logical violations flagged by deterministic rules, this analysis provides a comprehensive evaluation of detection performance.

2.6 Synthesis and Implications for System Design

The review of the state of the art reveals a divergence between conceptual and established models, evident in the literature and in the practical reality of managing public electric mobility. While the literature establishes individual standards for DTs, Anomaly Detection, and Visualisation, a unified platform capable of operating within the constraints of public networks remains unaddressed. This gap highlights four critical design drivers for this work:

1. **Operational Scope (Digital Shadow):** As identified in Section 2.1, the administrative restrictions of public infrastructure prevent the bidirectional control required for a full DT. Consequently, this thesis adopts a **DS**, supported by a **Microservices** architecture to ensure

the scalability required to ingest heterogeneous data streams without the rigidity of monolithic structures.

2. **Complexity Reduction via Visualisation:** As detailed in Section 2.2, the literature emphasises that a DS is only effective if its data is interpretable. Raw algorithmic output provides little value to non-expert operators. Therefore, the integration of **Geospatial Heatmaps** and interactive dashboards is not merely a cosmetic addition but a functional requirement for **Complexity Reduction**, transforming raw statistical indicators into actionable spatial insights.

3. **The Data "Blind Spot":** A critical limitation exists in standard detection methods for "Idle Occupancy." Crucially, within the specific public infrastructure integrated into this research, access to granular energy metrics is restricted. This creates an operational blind spot, necessitating a shift from telemetry-based detection to **Behavioural Profiling**, deriving anomalies solely from session duration and temporal patterns.

4. **Algorithmic Limitations:** The review of Unsupervised Learning identifies inherent limitations in algorithms often used in the field. K-Means is validated for segmentation but lacks specific outlier scoring, while Isolation Forest is robust for detection but relies on static “Contamination Rates” that introduce human bias. This functional gap underscores the need for an **Integrated Solution**, one that leverages profiling to calibrate detection sensitivity and address the lack of ground truth labels in public infrastructure data.

These findings form the basis for developing the platform. The following chapter, Proposed Solution, translates these theoretical requirements into a concrete functional design.

3 Proposed Solution

Building upon the theoretical foundations and technological standards established in the Related Work, this section presents the conceptual design and a high-level architectural methodology for the proposed DS platform. While the Related Works section served as the knowledge bridge connecting the identified limitations presented in the Introduction to the architectural decisions proposed here, this section acts as the design phase preceding the platform's concrete implementation. By translating theoretical concepts into specific requirements, it establishes a comprehensive blueprint of functional capabilities and performance standards that will guide the system's development.

To address this, the section is structured into four main segments. It begins by establishing the **Conceptual Vision**, providing a high-level abstraction of the ecosystem and the interaction between the physical assets and the digital platform. Building on this context, it specifies the **Functional Requirements** and the Platform's capabilities. The discussion then turns to the **NFR**, emphasising the system-wide quality metrics which established measurable criteria for the platform's technical implementation. The section concludes by presenting the **Architectural Overview**, introducing it in a high-level manner, providing a foundation for the detailed implementation that follows.

3.1 Conceptual Vision: The Digital Shadow Ecosystem

Prior to delineating specific functionalities, it is essential to establish the high-level conceptual model of the proposed system. The solution is designed as a **DS** of the public electric mobility network in Funchal, operating as a centralised intelligence layer that connects physical infrastructure to operational decision-makers, as illustrated in Figure 13.

At the base of this hierarchy lies the **Physical Layer**, composed of the distributed network of EVSE. These assets, specifically the physical chargers located throughout the municipality, represent the "Physical Twin." They generate continuous operational data, including status changes (e.g., Available, Charging, Out-of-Order) and session metrics (such as duration and energy consumption). However, given their status as public infrastructure integrated into the Mobi.e network, these assets do not offer Bi-directional control, restricting the system's scope to a DS rather than a full DT.

Connecting the physical and digital worlds is the **Data Ingestion Layer**. Since the municipality does not possess direct hardware access to the chargers' firmware, the platform relies on Mobi.e´s API integration to continuously mirror the state of the physical assets. This unidirectional data flow ensures that every change in the real world is propagated to the virtual environment.

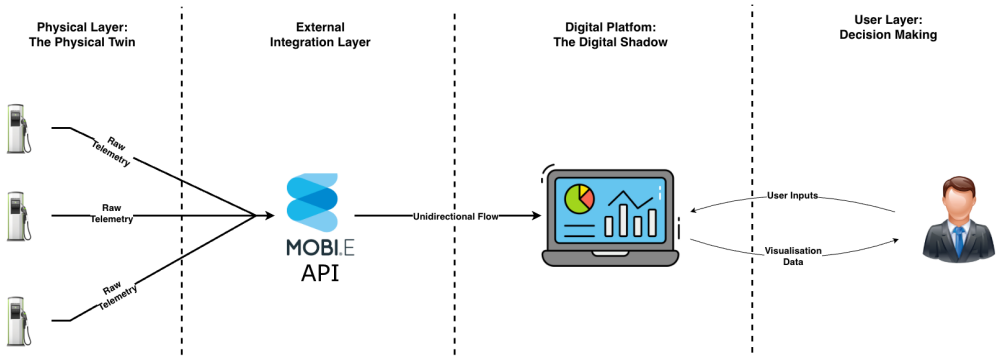


Figure. 13: Conceptual Vision of the DS Ecosystem, illustrating the data flow from physical assets to the decision-making layer.

The core of the system is the **Digital Platform**, which reconstructs this data to maintain a real-time virtual state of the network. More just simple mirroring, this layer supplements the raw data with analytical processing, such as calculating patterns, generating heatmaps, and detecting possible anomalies using algorithms. Finally, these insights are consumed by the **User Layer**, specifically infrastructure managers and operators, who utilise the platform to gain visibility into the network's health and optimise management strategies.

3.2 System Capabilities and Functional Requirements

With the high-level vision established, the platform is designed to deliver a selection of features, addressing specific user needs within the EV's and EVSE's charging ecosystem. These features are formalised as functional requirements, precise specifications that define what the system must do to fulfil its designated purpose.

To ensure these requirements reflected practical utility, they were established through direct engagement with the primary stakeholder, the CMF. Through ongoing project meetings, insights were gathered regarding the municipality's operational workflows and its specific management constraints. Based on these discussions, visual mock-ups of the proposed dashboard were developed. These prototypes were subsequently presented to CMF officials, confirming that the proposed functional assumptions accurately addressed their operational needs before finalising the system's

specifications. This process ensured that the final requirements effectively translate real-world challenges into practical, actionable features.

(The complete and formal specification of these functional requirements is detailed in Appendix A).

The core system functionalities derived from this collaborative process are detailed below.

3.2.1 Real-Time EVSE Infrastructure Monitoring

A fundamental requirement of the system is to provide a live, interactive and geographically-referenced visualisation of the entire charging network. The purpose of this capability is to deliver infrastructure managers and network operators immediate, high-fidelity operational awareness, being designed not as a public interface for EV drivers, but rather as a centralised monitoring hub, enabling the EVSE owners to monitor the health, real-time status and every specification of each one of the CS through a unified command interface. For operators, this translates into the ability to see beyond basic availability states (available, in-use, or non-operational) to include real-time data such as ongoing-charging time and system alerts indicating possible **"abusive parking"**.

This live monitoring represents one of the foundational elements of the DS, by creating a direct virtual link to the physical world, in this case, the real chargers which provide the data used in this platform. By providing a Single Source of Truth (SSOT) for the entire charging network's status, operational ambiguity is eliminated, ensuring the virtual model is a faithful representation of the physical world. This allows the EVSE owners and their operators to continuously monitor the network's health and validate operational performance against expected benchmarks, all presented within a unified, real-time view.

3.2.2 Data Analytics Dashboard

Complementing the real-time view, the platform must provide a robust analytical layer to facilitate the comprehension of historical and operational data. The purpose of this capability is to serve as the bridge between raw data streams and contextually significant information. The system must autonomously gather charging records from the Mobi.e API, standardise disparate data points into structured "Charging Sessions," and retain this information for long-term analysis.

Crucially, this data must be presented to the user through a suite of visual tools, such as graphs, diagrams, and statistical summaries, designed to simplify the interpretation of complex metrics.

This requirement ensures that infrastructure managers can easily assess key performance indicators (KPIs), such as average session duration, energy throughput trends, and peak usage hours, without needing to interact with raw database logs. This service acts as the platform's performance monitoring hub, ensuring that the DS provides not just a snapshot of the "present condition", but a deep understanding of the "history".

3.2.3 Geospatial Heatmap Visualization

To complement the analytical and monitoring dimensions, the platform integrates a Geospatial Heatmap Visualisation feature for the strategic analysis of historical usage patterns. Instead of presenting raw numerical data, this service specifically aggregates and maps the spatial intensity of charger usage across Funchal. This visual format enables stakeholders to identify long-term trends and spatial demand distributions.

The primary function of this service is to generate a geospatial heatmap layer that can be projected over the main map interface. This visualisation method allows mobility managers to discern spatial disparities between zones of peak demand and underutilised areas. Integral to this utility is the support for dynamic temporal selection, allowing users to define specific observation periods. When a user specifies a period, the service retrieves the relevant aggregated usage data and transforms it into the required visual format, where specific coordinates are mapped to intensity values corresponding to usage levels.

To ensure these intensity values remain analytically relevant, the system avoids applying a static, global scale across the entire network. Instead, the visual gradient dynamically recalculates its minimum and maximum thresholds based on the specific subset of chargers present within the operator's active spatial filter or immediate map view. This dynamic spatial scaling guarantees that high-demand hubs in dense urban centres do not disproportionately compress the visual representation when an operator is evaluating a different, lower-demand region.

By adapting to the user's immediate geographical context, this functionality supports the assessment of the current performance of the network and its operational management. By visually representing complex historical data, the service provides clear insights into usage patterns, identifies congested or underutilised assets, and assists in optimising operational strategies.

3.2.4 Anomaly Detection and Profiling

Finally, to transition the platform from a passive monitoring tool to an active decision-support system, the design incorporates a requirement for Data Profiling and Anomaly Detection. The objective of this capability is to automatically identify operational inefficiencies targeting "Idle Occupancy", an operational phenomenon that cannot be detected through simple deterministic thresholds due to the lack of real-time energy telemetry.

This requirement imposes the implementation of Unsupervised ML models capable of ingesting historical activity records datasets and constructing "Standard Usage Profiles". By grouping sessions based on duration and frequency, the system shall establish a statistical baseline of "normal user behaviour". Once this baseline is defined, the model must be able to autonomously flag sessions that deviate significantly from these patterns. These flagged events, representing potential anomalies or abusive behaviours, are immediately exposed to the operator, providing a high-value diagnosis that goes beyond simple state observation. This capability serves as the analytical layer of the DS, enabling the detection of invisible inefficiencies in the physical infrastructure.

3.3 Non-Functional Requirements

While the functional requirements define what the system is capable of and must do, the non-functional requirements (NFRs) specify how well it must perform these functions. These quality attributes and operational constraints are critical as they directly drive the architectural decisions detailed in the subsequent section (5. Implementation of the Digital Shadow Platform) and given the current pre-deployment phase of the project, these NFRs function primarily as foundational design principles rather than strictly measurable performance metrics. They are particularly important for this platform, whose primary design goal is to make the process of visualising and interpreting complex network data as effortless as possible for the user. These requirements ensure the platform's architecture is structurally prepared to be robust, efficient, and maintainable upon a future real-world deployment, ultimately delivering an intuitive and reliable user experience. The primary non-functional requirements acting as architectural drivers for the public mobility platform are outlined below.

Scalability: A primary non-functional requirement is the system's ability to scale. This encompasses both the capacity to manage increasing data volumes, such as high-frequency real-time inputs and large historical datasets, and the architectural flexibility to integrate new analytical

services over time. Since a DS increases its level of fidelity through the progressive incorporation of additional and more advanced models, the system must be designed to accommodate this functional growth without compromising the performance of its established capabilities.

Performance and Responsiveness: Given the platform's emphasis on real-time monitoring, low-latency data processing is a critical requirement. The credibility of a DS is fundamentally tied to its ability to act as a live, faithful mirror of the physical world, which necessitates that data be processed and delivered with minimal delay. This is particularly crucial for the Real-Time EVSE Infrastructure Monitoring service, whose architecture must be optimised to serve status updates instantly. Consequently, the backend services must be designed for high throughput, and the Graphical User Interface (GUI) must remain fluid and responsive. This responsiveness must be maintained even when visualising computationally intensive geospatial data layers, such as those generated by the Heatmap Service, or when handling high-frequency updates, ensuring a seamless and effective user experience.

Modularity and Maintainability: The platform's architecture must be essentially modular, in order to support its long-term evolution. As a DS is not a static system, its value increases as more models and services are integrated to enhance its representation of reality. Consequently, individual components and services must be designed as independent, autonomous units (interacting with minimal dependencies on each other). This is essential to facilitate parallel development and independent deployment, allowing, for instance, an update to a certain service without impacting or requiring a redeployment of another. Such a design also facilitates the future integration of new capabilities and is fundamental to ensuring the system can be updated and maintained over the long term without requiring a major architectural redesign.

Reliability and Availability: The system must be engineered for high reliability and availability, particularly for its core operational functions. The platform is required to be resilient to partial failures, ensuring that the failure of one service does not impact the availability of others, preventing a complete system-wide blackout. This means a failure within a single, non-essential analytical component must not impact the availability of other independent services. For instance, a failure event in an external data provider for one service should not impede an operator from using the functionalities of another, unrelated service. This principle of fault

isolation therefore strongly shapes the architectural design, which is essential for ensuring the overall robustness of the platform.

Usability: To ensure successful adoption by its designated stakeholders, such as infrastructure managers and network operators, the platform must be highly usable. The primary usability challenge is to present complex data across real-time, historical, geospatial, and temporal dimensions in an intuitive manner. The interface must be intuitively designed to minimise the cognitive load on the operator, translating complex datasets into clear visualisations. This requires a user experience that demands minimal specialised technical expertise, ultimately enabling a seamless transition from data to actual, valuable and interpretable information, allowing operators to achieve their goals efficiently.

3.4 Overall System Architecture: A Microservice-Based Approach

The architectural design of the platform is a direct response to the functional and non-functional requirements detailed previously. To meet these demands, a Distributed Microservices Architecture was selected. This strategic choice provides the foundation for a system that is designed to be scalable, resilient, and maintainable, directly addressing the core principles outlined in Section 3.3. To determine the optimal structure, two distinct architectural patterns were evaluated, a Monolithic Architecture and a Distributed Microservices Architecture.

3.4.1 Discarded Option: Monolithic Architecture

A traditional monolithic architecture, illustrated in Figure 14, was initially considered as the baseline approach. In this model, all functional modules, including data ingestion, analytics, user management and the interface logic, would be tightly integrated into a single, indivisible codebase.

However, this approach was eventually ruled out due to its structural limitations for this specific project. A monolithic approach would enforce a single, highly interdependent implementation across the system's diverse functionalities, removing the modular structure that provides the crucial conceptual separation between different functional domains and distinct areas of responsibility. Such an approach would create significant challenges related to maintainability and scalability, leading to a "dependency hell" that would impede the platform's long-term evolution.

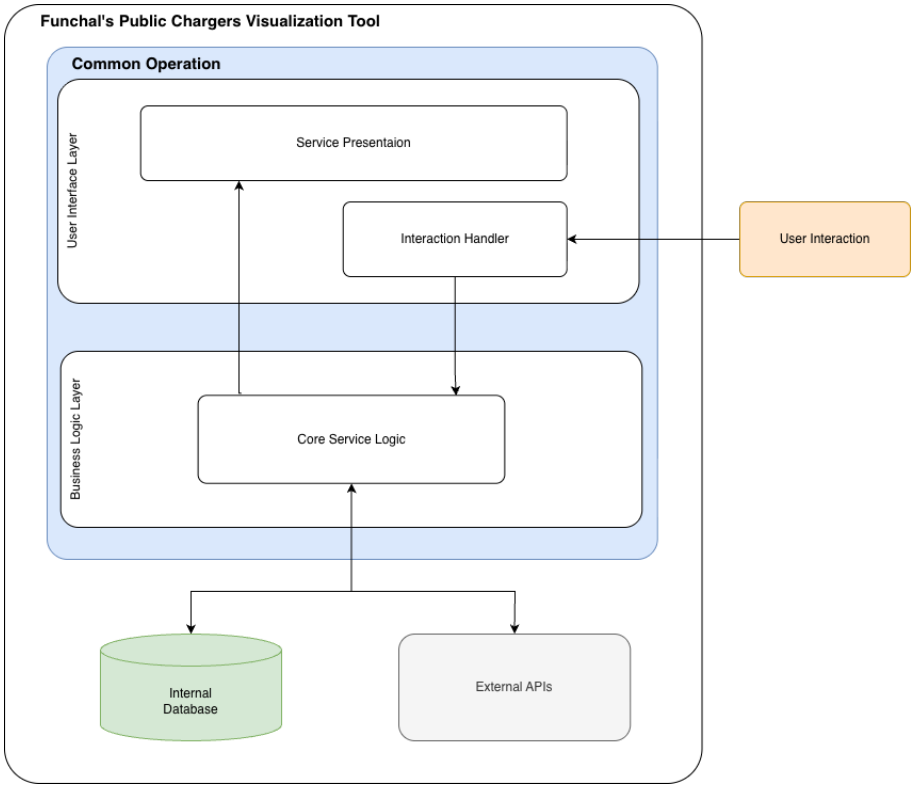


Figure. 14: High-level overview of the rejected Monolithic architectural approach.

3.4.2 Proposed Solution: Distributed Microservices Architecture

To overcome these limitations, a **Distributed Microservices Architecture** was selected as the final design. This strategic choice provides the foundation for a system that is scalable, resilient, and maintainable, directly addressing the core principles outlined in Section 3.3.

In contrast to the monolith, the microservice-based approach systematically decomposes the system into a collection of isolated, autonomous services, each responsible for a specific business capability. The high-level design of this chosen architecture is depicted in Figure 15.

Furthermore, this architecture, by its design, supports the non-functional requirements of Scalability and Reliability. Each Microservice can be scaled independently based on its specific resource demands. For instance, a service experiencing high demand can be allocated more computational resources independently, without impacting the performance of other services. This level of control over individual components ensures efficient resource utilisation and maintains *Performance and Responsiveness*.

From a reliability perspective, the architecture provides fault isolation. The failure of a single service, such as the *heatmap generation*, does not cascade to cause a system-wide "blackout". Other

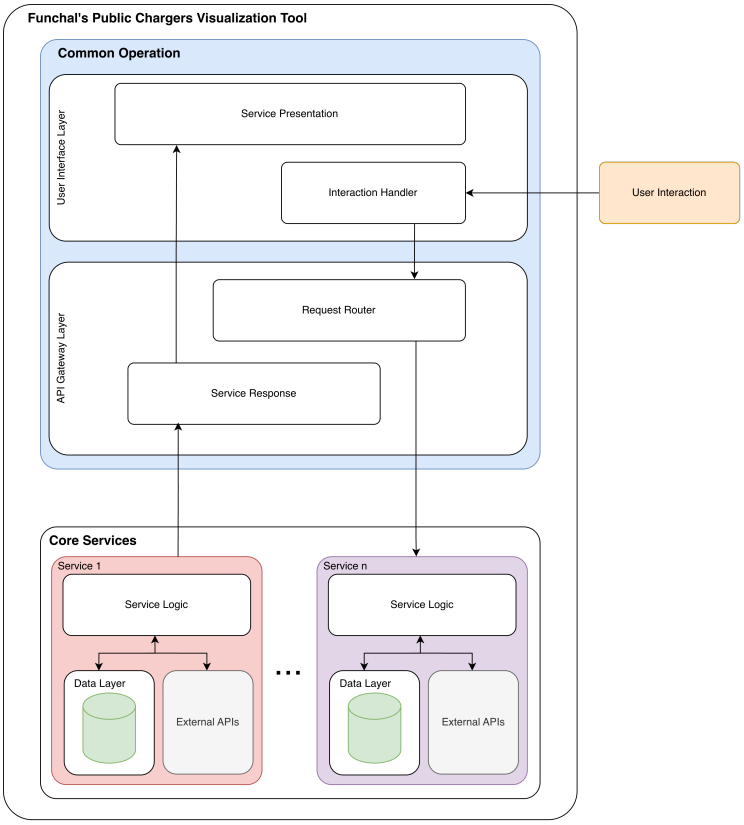


Figure. 15: High-Level Architecture of the proposed DS Platform, illustrating the Layered Microservices approach.

core functionalities, like *live map monitoring*, remain available, ensuring high Availability for the remaining operations.

To translate these architectural principles into a concrete and organised structure, the proposed architecture is arranged into two primary tiers: the **Platform Common Operation Layers**, which provide the unified user experience, and the **Backend Services Layer**, which contains the independent Microservices.

These tiers are further subdivided into three distinct layers, creating a clear and robust separation of concerns:

– **User Interface (UI) Layer:** This is the presentation layer, a client-side dynamic web interface responsible for all user interactions. It renders the map, visualises data, and provides the controls for interacting with the platform's features. It communicates with the backend exclusively through the API Gateway, ensuring a clean separation from the business logic and directly addressing the Usability requirement by focusing on providing a responsive and intuitive user experience.

- **API Gateway Layer:** Serving as the single entry point for all client requests, the API Gateway simplifies the system's external access points and acts as an orchestrator, routing incoming requests to the corresponding Microservice in the Backend Services Layer. This layer serves as a structural element of the architecture by abstracting the internal complexity of each service.
- **Backend Services Layer:** This layer contains the system's business logic, which is decomposed into a set of independent, single-purpose Microservices. Each service encapsulates a specific functional domain:
  - **Real-Time Map Service:** Responsible for ingesting and broadcasting real-time status updates from the EVSE network;
  - **Data Analytics Service:** Acts as the analytical processing engine, ingesting raw data from the Mobi.e API, formatting it into structured charging sessions, and delivering the required statistical metrics needed for data visualisation;
  - **Heatmap Service:** Operates as a high-volume analytical tool for historical usage visualisation;
  - **Anomaly Detection Service:** Performs as the algorithmic core of the platform, utilising the Unsupervised Learning models to detect operational anomalies.

This high-level architectural overview establishes the platform's structural foundation. The data strategies required to build the anomaly detection models will be defined in the "Data Profiling and Anomaly Detection Strategy" section, followed by the detailed technical implementation of the entire platform in the "Implementation of the Digital Shadow Platform" section.

4 Data Profiling and Anomaly Detection Strategy

This section details the analytical methodology used to transform raw charging records into operational insights by establishing the behavioural logic necessary to identify suspicious patterns within the dataset. The primary objective is to develop a reliable anomaly detection engine capable of identifying operational inefficiencies, particularly “Station Overstaying”, without relying exclusively on static rules.

However, the design of this engine is fundamentally shaped by the **operational constraints** identified in the problem statement. As highlighted in Section 2.6: Synthesis and Implications for System Design, the system operates with a “Data Blind Spot” due to the unavailability of real-time energy telemetry. Consequently, inefficiencies cannot be detected through direct power monitoring (e.g., flagging when power drops to zero), but must instead be inferred indirectly through behavioural analysis.

To achieve this, a **Hybrid Unsupervised Learning approach** was adopted, combining the **macro-level profiling** capabilities of **K-Means Clustering** with the **local outlier detection** precision of the **Isolation Forest** algorithm. The methodology follows a linear pipeline, represented in Figure 16. It begins by ingesting and transforming historical data into feature vectors, allowing K-Means to establish a baseline, differentiating “normal” from “suspicious” behaviour. This baseline subsequently calibrates the Isolation Forest model to detect specific anomalies. This synergy ensures the DS acts not just as a mirror of the physical network, but as a diagnostic tool.

4.1 Data Acquisition and Pre-processing

The foundation of the anomaly detection system is the "Charging Session" dataset. For the development of these models, this analysis utilises historical data files provided directly by the municipality in static spreadsheet formats, such as Excel files. Before this data can be consumed by machine learning models, it undergoes a sequential pre-processing mechanism of ingestion, cleaning, and mathematical transformation to ensure quality and consistency.

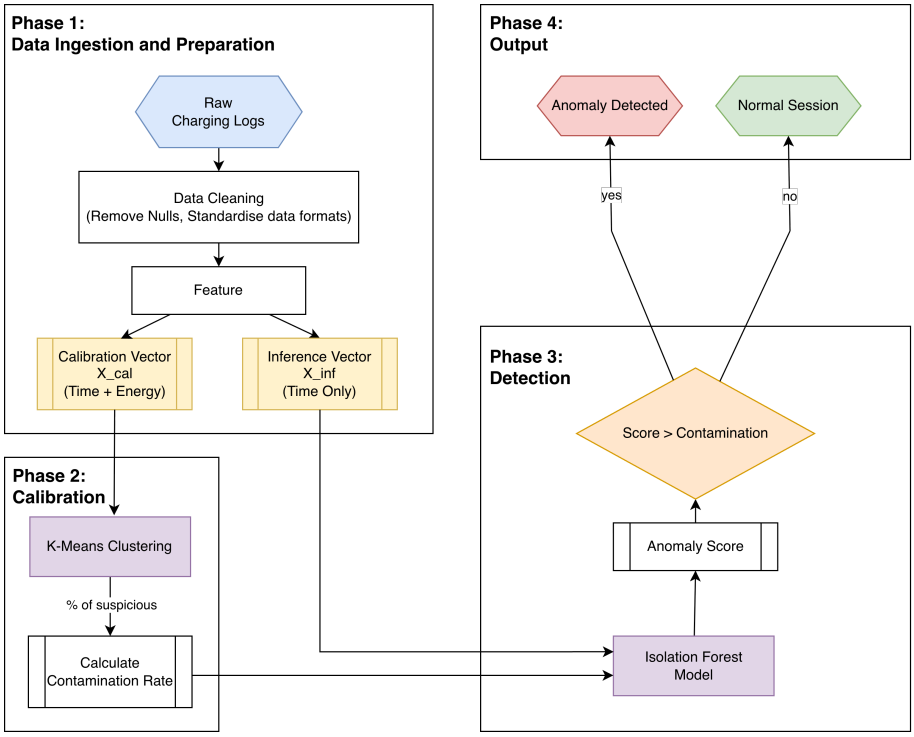


Figure. 16: Hybrid Anomaly Detection Pipeline: Data Preparation, Calibration (K-Means), and Detection (Isolation Forest)

4.2 Data Ingestion

The data pipeline begins with the physical ingestion of the historical charging records. These datasets aggregate charging records from previous months, serving as a representative sample of the network's operational history. The machine learning pipeline dynamically loads these static files into memory, utilising Pandas DataFrames, transforming the spreadsheets into a structured, programmable format.

At this initial point of entry, each record consists of strictly unprocessed transaction details directly mirroring the municipal exports. This includes raw string timestamps, categorical identification codes (such as EVSE IDs), and unverified energy metering values. Establishing this in-memory structural representation is a critical first step, ensuring that the pipeline has a secure and fully loaded dataset before any algorithmic cleaning begins.

4.2.1 Data Cleaning and Standardisation

Once the raw dataset is successfully ingested into the pipeline, the immediate step focuses on cleaning the data structure. The system removes records with critical missing values (e.g., null timestamps or undefined energy values) to prevent model corruption and analytical bias, and

timestamps originally formatted as strings (e.g., "YYYYMMDDHHMMSS") are parsed into Python `datetime` objects to enable temporal calculations.

4.2.2 Feature Engineering and Mathematical Derivations

To enable the algorithms to detect complex behavioural patterns, the raw logs are transformed into a feature representation that captures the physical characteristics of a charging session. The system generates two categories of features: **Temporal and Duration Features** (available for all sessions) and **Energy Efficiency Features** (available only when energy telemetry is present).

4.2.2.1 Temporal and Duration Features

The first set of features focuses on temporal dimensions. By parsing the raw timestamps (t_{start}, t_{stop}), the system calculates the precise **Session Duration** (Δt) in minutes:

$$\Delta t = \frac{(t_{stop} - t_{start})}{60} \tag{1}$$

Additionally, temporal and seasonal features are extracted to capture usage patterns:

- **Start Hour (H):** The hour of the day when the session began, $H \in [0, 23]$, enabling the model to distinguish between peak-hour usage and overnight parking.
- **Day of Week (D):** A numerical index $D \in [0, 6]$ where 0=Monday and 6=Sunday, used to distinguish weekday to weekend usage patterns.

Unlike energy metrics, these temporal dimensions are guaranteed to be present in every session analysed. Consequently, for sessions where energy data is missing, they transition from being supportive metrics to becoming the primary and exclusive determinants of the risk score.

4.2.2.2 Energy Efficiency Features

For instances where consumption metrics are available, the system extracts derived metrics specifically designed to flag "abusive" behaviours. The primary metric is the **Energy Efficiency** (η), representing the average power delivery rate:

$$\eta = \frac{E_{total}}{\Delta t} \quad (\text{kWh/min}) \tag{2}$$

This metric is critical for identifying "Station Overstaying". A standard electric vehicle actively charging will maintain a high η. In contrast, a vehicle that has finished charging but remains occupying the station will have its E_{total} static while Δt increases, causing η to tend to zero.

To formalise this detection, a *Low Energy Flag* (F_{low}) is mathematically defined as a binary indicator:

$$F_{low} = \begin{cases} 1 & \text{if } \eta < 0.05 \text{ kWh/min} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Additionally, the system identifies **Phantom Sessions** as sessions in which a connection failure or an immediate disconnect occurred. This is detected through a binary flag ($F_{phantom}$) defined as:

$$F_{phantom} = \begin{cases} 1 & \text{if } (E_{total} < 1 \text{ kWh}) \wedge (\Delta t > 60 \text{ min}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

4.2.2.3 Feature Summary and Normalisation

With both the temporal and energy-based metrics formally established, the system initially consolidates them into a comprehensive feature definition. This integration is essential, as it provides a multidimensional view that contextualises energy usage against time-based variables.

However, to align with the operational constraints identified in the problem statement (specifically, the lack of real-time energy telemetry), this feature set is methodologically divided into two distinct vectors:

- **Calibration Vector (X_{cal}):** Includes both *Temporal* and *Energy* features. This full-fidelity vector is used exclusively during the offline training phase by the **Rule-Based Analysis** to establish the analytical reference, and by the **K-Means algorithm** to accurately identify "Station Overstay" patterns and derive the system's baseline parameters.

- **Inference Vector (X_{inf}):** Includes only *Temporal* features. This reduced vector is used by the **Isolation Forest** to detect anomalies in real-time, ensuring the system functions even when energy data is unavailable.

The complete selection of features and their allocation to the respective training and detection stages is summarised in Table 1.

Feature Source	**Variable**	**Calibration (X_{cal})**	**Inference (X_{inf})**
Temporal Parsing	`start_hour`	Yes	Yes
	`start_day_of_week`	Yes	Yes
	`calculated_duration`	Yes	Yes
Energy Metrics	`energy_efficiency` (η)	Yes	No
	`total_energy` (E_{total})	Yes	No
	`low_energy_flag` (F_{low})	Yes	No
	`phantom_flag` ($F_{phantom}$)	Yes	No

Table 1: Feature Allocation for Training and Detection Stages

Finally, to prevent features with large magnitudes (e.g., Duration in minutes) from dominating features with smaller ranges (such as Efficiency in kWh/min) during the distance calculation in clustering, all numerical input vectors (X_{cal} and X_{inf}) are normalised using **Standard Scaling**, resulting in a Z-score distribution:

$$z = \frac{x - \mu}{\sigma} \tag{5}$$

Where:

- x is the original feature value (e.g., raw duration).
- μ is the **Average** value for that feature across all history (e.g., the average session is 120 minutes).
  Subtracting the average value from the original centres the data around zero.
- σ is the **Standard Deviation**, representing how much the data usually varies.
  Dividing by this removes the unit (e.g., minutes/kWh), putting all features on the same "ruler."

This mathematical step is crucial, especially because K-Means uses spatial distance to determine similarity. Without scaling, a variable like duration (0-600) would disproportionately outweigh efficiency (0-1), leading the algorithm to ignore the smaller-scale feature purely due to the difference in units.

4.3 Baseline Strategy: Rule-Based Analysis

Before applying unsupervised machine learning techniques, a deterministic baseline is established using a **Rule-Based Physics Analysis** that does not "learn" patterns from data distributions but instead relies on the immutable physical laws of energy transfer to calculate the theoretical minimum time required for a charging session. This serves two purposes, as it provides an explainable "deterministic proxy" for clear-cut abuse cases and acts as a standard for comparing the performance of the subsequent ML models.

4.3.1 Theoretical Overstay Calculation

The core challenge in identifying "Station Overstay" (parking after charging is complete) is that the specific charging capability ($P_{vehicle}$) of the connected electric vehicle is unknown to the station. However, for any given power rate scenario, the theoretical necessary charging time (T_{est}) can be calculated as:

$$T_{est} = \left(\frac{E_{total}}{P_{scenario}} \right) \times 60 \quad \text{(minutes)} \tag{6}$$

Where E_{total} is the total energy consumed (kWh) and $P_{scenario}$ is the assumed power rate (kW). Consequently, the "Abusive Duration" (T_{abuse}) is defined as the time the vehicle occupied the station beyond this necessity:

$$T_{abuse} = T_{total} - T_{est} \tag{7}$$

To quantify the severity of the violation, we define the **Abusive Rate** (R_{abuse}) established by Biswas et al. [35], which represents the proportion of the session spent idling:

$$R_{abuse} = \frac{T_{abuse}}{T_{total}} \times 100 \tag{8}$$

4.3.2 Multi-Scenario Sensitivity Analysis

Since the actual vehicle type and its maximum intake capacity (P_{max}) represent a latent variable unknown to the charging station, applying a single reference parameter would lead to inaccurate results. To resolve this ambiguity, the system defines three distinct charging scenarios derived from

the reported characteristics of the Portuguese EV fleet and the physical specifications of the local infrastructure.

4.3.2.1 User-Driven Capacity Distribution

To ensure the analysis reflects the reality of local drivers, data from the study conducted by the **Automóvel Club de Portugal (ACP)** regarding EV user behaviour and hardware specifications was integrated [45]. The survey captured the diversity of the national fleet, providing valuable insights into the charging limitations users face. As shown in Table 2, the distribution of reported capacities highlights a highly fragmented market.

Charger Power Range (kW)	Percentage of Respondents
Less than 7 kW	5%
From 7 kW to 11 kW	12%
From 11 kW to 22 kW	17%
From 22 kW to 50 kW	13%
From 50 kW to 150 kW	7%
More than 150 kW	2%
Does not know	44%

Table 2: Distribution of Reported EV Charger Capacities in Portugal

4.3.2.2 Infrastructure Operational Context

Although the study identifies vehicles with intake capacities exceeding 50 kW, the local charging ecosystem is constrained by the specifications of specific municipal chargers, which impose a standardised maximum output of 22 kW per charger. This configuration balances service availability with grid stability. Consequently, regardless of a vehicle's theoretical maximum, the 22 kW ceiling of the physical station serves as the effective upper bound for this analysis.

4.3.2.3 Defined Analysis Scenarios

By combining the user data from the Portuguese fleet with these operational constraints, the system evaluates every session against three specific scenarios:

– **Scenario A (22 kW) – Maximum Infrastructure Capacity:** Represents the peak power available from the EVSE. This is the baseline for high-capacity vehicles when they have exclusive access to the charger's output.

– **Scenario B (11 kW) – Shared Infrastructure:** When two vehicles utilise the same double-connector EVSE, the 22 kW supply is typically divided, giving 11kW for each. Additionally, this limit also aligns directly with the largest identifiable segment of the study's population, effectively bridging infrastructure constraints with vehicle capabilities.

– **Scenario C (7 kW) – Fleet Minimum:** Derived from a lower but still relevant percentage of users in the Portuguese study reporting capacities along the 7 kW range. This conservative scenario ensures that older EVs are also taken into consideration.

4.3.2.4 Consensus Mechanism for Abuse Detection

A session is flagged as "Abusive" within a specific scenario if $R_{abuse} \geq 50\%$. The final classification relies on a consensus of these three viewpoints:

– **High-Confidence Abuse:** Flagged in **ALL** scenarios. Even under the most conservative assumptions (e.g., assuming the slowest possible charging speed of 7 kW), the user occupied the station significantly longer than necessary.

– **Probable Abuse:** Flagged in 2 out of 3 scenarios.

– **No Abuse / Ambiguous:** Flagged in 0–1 scenarios. These cases are treated as legitimate charging sessions.

4.4 Modelling Strategy: K-Means Clustering

The Rule-Based Physics Analysis provides a necessary "heuristic reference" for enforcement, yet it remains a rigid diagnostic tool restricted by static thresholds. It identifies *known* types of abuse but is blind to emerging irregularities or complex usage patterns that do not strictly violate the specific duration-energy formulas.

To capture these non-obvious usage variances, the system employs **K-Means Clustering** as a profiling engine. Crucially, consistent with the feature strategy defined in Section 4.2.2.3, this algorithm is applied in two distinct operational configurations:

1. **Calibration Profiler (using X_{cal}):** Utilises the full feature set (including energy) to accurately identify "Station Overstay" patterns during the offline training phase. This establishes the empirical baseline used to calibrate the Isolation Forest.

2. **Operational Profiler (using X_{inf}):** Utilises only temporal features to segment live data. This instance provides the real-time behavioural context required for the Consensus Logic (Section 4.6.2).

By utilising these normalised feature sets established in Section 4.1, the algorithm enables the system to move from *checking* for violations to *discovering* behavioural profiles. This allows the definitions of "normal" and "suspicious" activity to be derived directly from the statistical structure of the network's usage history rather than external presuppositions.

4.4.1 Theoretical Foundation

K-Means is a centroid-based algorithm that segments the dataset $X = \{x_1, ..., x_n\}$ into k distinct clusters $C = \{C_1, ..., C_k\}$ by minimising the internal variance of each group, creating compact and homogeneous profiles. To achieve this, the algorithm relies on an iterative procedure that alternates between assigning data points to the nearest cluster and updating the cluster centres until the system reaches a stable state. This process relies on the objective function J, which serves as a quantitative 'score' of compactness. In this context, by minimising this value, the algorithm systematically reduces the total distance between the sessions and their assigned profiles. The objective function J is defined as:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - \mu_j||^2 \tag{9}$$

Where:

- J is the **Total Distortion** (or cost) that the algorithm tries to minimise.
- $\sum_{j=1}^{k} \sum_{i=1}^{n}$ represents the sum of the error for every single data point i inside every single cluster j.
- $x_i^{(j)}$ represents a specific charging session that belongs to that group.
- μ_j represents the **Centroid** of the cluster, the exact centre point of that group.
- $||...||^2$ calculates the Distance between the session and the centre of its group.

Conceptual Translation:

Fundamentally, this formula measures the total "dispersion" or variance within the different clusters/profiles. The algorithm iteratively moves the centroids (μ_j) to the mathematical centre of the data points, aiming to make each cluster as tight and compact as possible.

4.4.2 Implementation Strategy

To transition from the theoretical minimisation of J to an algorithmic execution, a structured processing pipeline is required. While the objective function J effectively drives the centroids to the centres of their respective groups during training, it cannot independently determine the optimal analysis resolution, as a pure minimisation of J alone would simply favour an infinite number of clusters. Therefore, the procedure is split into two phases. First, an external evaluation metric (Silhouette Analysis) is used to identify the structural 'sweet spot' (k) that balances the trade-off between internal tightness and external spacing between the clusters. Second, the model is trained using this optimal parameter.

4.4.2.1 Step 1: Determining Optimal k

Since K-Means requires the number of clusters (k) to be pre-determined, the implementation uses an iterative approach. While the optimal k is traditionally identified through the Elbow method, this approach relies heavily on subjective visual interpretation to find an inflection point. To enable an autonomous data pipeline, the system instead evaluates a candidate range of k values (from 2 to 10) using the **Silhouette Coefficient** (s). Unlike the Elbow method, this metric provides a precise mathematical score, measuring how similar an object is to its own cluster compared to other clusters, which allows the system to programmatically select the optimal configuration without human intervention.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{10}$$

Where:

- $s(i)$ is the **Silhouette Score** for a specific data point i, ranging from -1 (incorrect clustering) to +1 (perfect clustering).
- $a(i)$ measures **Cohesion**: The average distance from point i to all other points in its own cluster (Lower is better).
- $b(i)$ measures **Separation**: The average distance from point i to all points in the *nearest* neighbouring cluster (Higher is better).
- $\max\{a(i), b(i)\}$ acts as a **Normalisation Factor**, scaling the result so it always falls between -1 and 1.

To translate this into its implementation, the system adopts a comparative validation strategy. It iterates over a candidate range of k values (typically 2-10) and fits a distinct and temporary K-Means model for each k, resulting in the definition of stable centroids and distinct cluster structures. Within each iteration, utilising the specific cluster assignments derived from the temporary fit, the algorithm calculates the Silhouette Coefficient for every individual data point and computes the overall average score. Finally, the system compares these averages and selects the specific k that corresponds to the best value, effectively automating the discovery of the configuration where the optimal balance between internal compactness and external separation is optimised. (see Appendix B.1 for the logic optimal number of cluster).

4.4.2.2 Step 2: Model Fitting and Profile Characterisation

Once the optimal number of clusters (k) is determined through Silhouette Analysis, the final model is deployed to segment the dataset. This step executes the iterative centroid update process described by the objective function J, minimising the internal variance to reach a stable centroid state. The system then instantiates the model with the optimised parameters and assigns each historical session to a specific behavioural profile based on its spatial proximity to the nearest cluster centroid (see Appendix B.2 for the fitting implementation).

At the conclusion of the training phase for both the Calibration and Inference vectors, the model effectively segments the charging sessions into distinct behavioural groups. For this dataset, the analysis (typically selecting $k = 2$) consistently reveals two primary usage profiles, as detailed in Table 3:

By executing this workflow, the implementation practically implements the theoretical segmentation, clearly isolating 'Station Overstay' events from standard charging sessions based on their

Profile	Profile Description
Standard Charging	Defined by shorter durations (< 120 mins) and high energy efficiency (observable when using X_{cal}). These sessions represent legitimate usage patterns where the connection time correlates directly with the energy transferred.
Station Overstay *(Suspicious)*	Characterised by "excessive" durations (> 240 mins) combined with minimal energy efficiency (evident only through X_{cal}). These events frequently occur overnight or on weekends, indicating abusive station occupancy without active charging.

Table 3: Behavioural Profiles Identified by K-Means Clustering

duration and efficiency profiles. Consequently, the K-Means method fulfils its primary objective within this methodology, serving to define a preliminary reference baseline from unlabelled data, providing the reference labels required to calibrate and validate the following Anomaly Detection algorithm.

4.5 Anomaly Detection: Isolation Forest

While K-Means is excellent for profiling general behaviours, it forces every data point into a cluster, making it less effective at identifying unique edge cases. To address this, the platform implements the **Isolation Forest** algorithm specifically for outlier detection.

4.5.1 Theoretical Foundation

Unlike distance-based methods that rely on density, Isolation Forest relies on the principle that anomalies are "few and different". For this purpose, a forest of random binary decision trees (iTrees) is generated.

Instead of classifying or profiling the data, the algorithm uses a recursive random partitioning process. For every tree in the forest, the algorithm repeatedly splits the data by randomly selecting a feature and a cut-point value between the minimum and maximum range of that feature. This process continues recursively until every data point is isolated in its own leaf node.

The core intuition is that anomalies, which possess attribute values that differ significantly from the norm, are statistically likely to be separated from the rest of the data very early in this random process (requiring fewer cuts). Conversely, normal points are concentrated in dense regions with similar values. Due to the high concentration of proximal instances, a single random cut rarely

isolates them. Instead, the algorithm needs to make many successive cuts to gradually separate the specific point from the crowd.

The final anomaly score $s(x, n)$ is derived by comparing the specific depth of a point against the statistical average:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{11}$$

Where:

- $s(x, n)$ is the final **Anomaly Score** for instance x, bounded between 0 and 1.
- $2^{-(\dots)}$ is an exponential decay function. It ensures that as the path length $E(h(x))$ decreases, the score approaches 1 (Anomaly). Conversely, as the path length increases, the score drops towards 0 (Normal).
- $E(h(x))$ is the **Average Path Length**: The number of random cuts required to isolate the specific session x, averaged across the entire forest (e.g., 100 trees). This approach ensures statistical stability, mitigating the influence of "luck" on the final score.
- $c(n)$ is the **Baseline Normalisation**: The average path length of a generic Binary Search Tree for a dataset of size n.

Conceptual Translation:

Fundamentally, this formula measures the structural 'isolation depth' of a data point relative to the rest of the dataset. The algorithm applies random cuts to the data, aiming to distinguish anomalies, which are isolated very early, if compared to regular points that require a long sequence of cuts to finally reach a leaf node.

4.5.2 Implementation Strategy

The implementation applies the theoretical concept by initially creating a forest of randomised trees to compute anomaly scores for each data point using their path-length metrics.

However, a key distinction must be made between theory and practice. While the theoretical formula produces a continuous score $s(x, n)$ between 0 and 1, it does not intrinsically define an "explicit decision value" for what constitutes an anomaly. To resolve this classification ambiguity, the implementation introduces a specific cut-off value, the `Contamination Rate`, for binary predictions (Normal vs. Anomaly).

4.5.2.1 Step 1: Model Initialisation and Thresholding

The implementation process begins by configuring the algorithm's structural parameters within the feature space defined by the **Inference Vector (X_{inf})**. To execute the binary classification logic, the implementation utilises the `contamination` parameter as the primary control mechanism.

Functionally, this acts as a **rank-based constraint** rather than an absolute threshold. Instead of evaluating whether a score exceeds a fixed value (e.g., $s(x, n) > 0.7$), the algorithm calculates scores for the entire dataset, sorts them, and is forced to designate the top p percent (defined by the contamination rate) as anomalies. This implies that the model's sensitivity is relative: by setting `contamination=0.1`, the system is constrained to flag the "worst" 10% of sessions, regardless of whether the dataset is clean or highly polluted.

To ensure this sorting process is reliable, the model structure is supported by the `n_estimators` parameter. As established in the theoretical foundation, a single tree is susceptible to random noise. Therefore, the implementation generates a forest of independent trees to mitigate these variations, ensuring that the anomaly scores used by the contamination parameter are consistent and reliable.

For this specific implementation, the model is configured with the following configuration variables to define its operational limits:

- **Contamination Rate:** Established at a default value of 0.1 (10%), this parameter serves as the explicit instruction to identify the highest-scoring segment (corresponding to the shortest path lengths) as anomalies.

- **Estimators (n_estimators = 100):** Configured with 100 independent trees, the model meets the theoretical requirement for a forest structure, mitigating the variance of random partitioning and ensuring the stability of the final path length $E(h(x))$.

Once the parameters have been initialised, the system configures the model with the previously defined contamination rate, effectively establishing the cut-off criteria that will be applied to future scores. With this classification rule set, the final step is to build the forest. The *.fit()* method processes the dataset and physically constructs the 100 trees in memory. This establishes the reference model used to evaluate all charging sessions (see Appendix B.3 for the initialisation details).

4.5.2.2 Step 2: Scoring and Prediction

Once the forest structure is fully established (fitted), the algorithm transitions to the scoring phase. It propagates each session down the hierarchy of the 100 trees, calculating the average path length required to isolate each data point. This depth measurement serves as the raw anomaly score, which is subsequently compared with the contamination parameter's cut-off limit to produce the final binary prediction (as detailed in Appendix B.4).

Note: While effective for fixed datasets, this standard implementation relies on a static contamination assumption. Consequently, it lacks the ability to automatically adapt if the prevalence of abuse changes over time. This specific limitation is addressed in the subsequent ***Hybrid Strategy****.*

4.6 Anomaly Detection: Hybrid Strategy

Building upon the limitations identified in Sub-Section 4.5: Anomaly Detection: Isolation Forest, the standard Isolation Forest is constrained by the requirement for a manually selected *Contamination Rate*.

Relying on **arbitrary estimation** for this critical parameter introduces subjective bias and operational risk. For example, if the model is configured with a fixed 10% contamination rate, but the actual abuse frequency in the network rises to 20%, the system will **systematically fail to detect half of the anomalies**. Conversely, setting the rate too high results in excessive false alarms, eroding trust in the system.

To minimise the reliance on this dependency, the platform employs a **Hybrid Approach** that integrates K-Means and Isolation Forests into a **self-calibrating pipeline**.

4.6.1 Data-Driven Calibration Strategy

In this architecture, the models operate in a hierarchical dependency, where **K-Means** serves as the *Calibration Agent* and **Isolation Forest** as the *Detection Engine*, each one operating on distinct feature configurations. While standard hybrid models typically use the same features for both steps, this approach utilises the **Calibration Vector** (X_{cal}) to establish the reference baseline and the **Inference Vector** (X_{inf}) to perform the detection. The pipeline executes in two stages:

1. Calibration Phase (K-Means on X_{cal}): The K-Means algorithm initiates the process by profiling the historical dataset using the full **Calibration Vector** (including Energy), with the

intention of identifying the "Station Overstay" cluster (characterised by long durations and low energy). Based on the proportion of this specific group within the total population, the system quantifies the empirical probability of abuse, $P(C_{suspicious})$.

2. Detection Phase (Isolation Forest on X_{inf}): This empirical probability, $P(C_{suspicious})$, is then used to mathematically derive the dynamic *Contamination Rate* (C_{rate}) to be applied within the Isolation Forest algorithm. Crucially, this model is configured with this rate but derived solely from the **Inference Vector** (**temporal data**). Effectively, the energy data is used to "instruct" the temporal model how sensitive it should be.

However, to enforce a conservative detection limit and minimise possible false positives, a **Confidence Adjustment Factor** ($\alpha = 0.85$) is applied:

$$C_{rate} = P(C_{suspicious}) \times \alpha \tag{12}$$

Where:

- C_{rate} is the final **Contamination Parameter** passed to the Isolation Forest, representing the expected percentage of outliers.
- $P(C_{suspicious})$ is the **Empirical Probability** of abuse, derived from the percentage of total sessions that K-Means assigned to the "Suspicious" cluster.
- α is the **Confidence Adjustment Factor** (set to 0.85). It acts as a safety buffer, assuming that even within a suspicious cluster, roughly 15% of sessions might be legitimate atypical cases, consequently reducing false positives.

Using this logic ensures that the outlier-detection algorithm adapts its sensitivity to the actual observable incidence of abuse in the network's history, rather than arbitrary guesses. The programmatic implementation of this sequential calibration process first identifies the cluster with the highest average duration as the "Suspicious" baseline. It then calculates the proportion of total sessions in this group to derive the raw abuse rate, and finally applies the safety buffer (α) to compute the dynamic contamination parameter (see Appendix B.5 for the calibration algorithm).

4.6.2 Consensus Logic

As the system enters the inference stage, the integration of the presented models expands beyond the calibration of the *Contamination Rate* to become the core of the risk assessment logic. To ensure the veracity of the detected anomalies, the system applies a **dual-validation logic** during

the detection cycle. This approach overlays the independent outputs of both the behavioural profiler and the detection algorithm, requiring them to converge on a decision to fully validate a threat.

To operationalise this convergence, for any individual session, is derived from two parallel evaluations, both utilising Inference Vector (X_{inf}):

1. **Contextual Classification (K-Means):** The system determines the session's behavioural context. It calculates the spatial proximity to the existing centroids of a temporal baseline model and assigns the session to the closest cluster ("Normal" or "Suspicious").
2. **Statistical Scoring (Isolation Forest):** Simultaneously, the system evaluates distributional irregularity by calculating the candidate session's anomaly score and comparing it against the self-calibrating limit (C_{rate}) established during the calibration process. If the score exceeds this limit, the session is flagged as an anomaly.

The final decision is not binary but a **superposition** of these two signals. By aligning the behavioural context with the statistical score, the system assigns a multi-contextualised "Risk Level" as detailed in Table 4:

K-Means Output	Isolation Forest Output	Risk Level	System Action
Suspicious Cluster	Anomaly (-1)	**HIGH**	Automated Alert
Normal Cluster	Anomaly (-1)	**MEDIUM**	Manual Review
Suspicious Cluster	Normal (1)	**MEDIUM**	Manual Review
Normal Cluster	Normal (1)	**LOW**	No Action

Table 4: Hybrid Consensus Matrix for Anomaly Risk Classification

4.6.2.1 Real-Time Inference via Digital Shadow

Implementing the logic defined above, the DS applies this matrix as its primary decision engine when processing new, real-world sessions. From a structural perspective, this logic is encapsulated within the **Anomaly Detection Service**, operating as an external Microservice accessible via the **API Gateway**, as defined in the system architecture.

Crucially, this service operates exclusively in its **inference mode**, as it does not re-train the algorithms dynamically but rather utilises the static model states generated by the calibration pipeline. Following the arrival of a live data stream via the API, the system projects the new session

against these established baselines (the K-Means centroids and the Isolation Forest structure). Consequently, by applying the logic defined in the **Consensus Matrix** to these pre-calculated references, the DS is able to assess the risk level of incoming sessions relative to the calibrated baselines established by the reference models.

5 Implementation of the Digital Shadow Platform

Following the conceptual design and architectural decisions outlined in Chapter 3, this chapter details the concrete technical realisation of the DS platform. While the previous section 3 defined *what* the system should do, this section explains *how* it was engineered to achieve those goals.

The implementation relies on a **Layered Microservices Architecture** established in Section 3.4, a choice driven by the need to isolate complex data processing tasks from the user interface. By decoupling these components, the system ensures that heavy computational tasks, such as processing thousands of charging records or running machine learning models, do not compromise the responsiveness of the interactive map. Before detailing the architectural layers, the following subsection summarises the main technologies used in its implementation.

5.1 Technology Stack

The implementation adopts different technologies according to each service concern. The API Gateway and the main backend services are implemented in **Node.js** using ES modules with **Express** providing HTTP endpoints and routing logic. In the gateway path, requests and responses are forwarded as streams, allowing data to flow progressively instead of being fully stored in memory first, reducing memory usage and response times.

Persistent storage and analytical workloads rely on **PostgreSQL**, accessed through the Node.js `pg` client. This setup allows filtering, aggregation, and indexed queries to be executed in-database, reducing payload size and avoiding unnecessary processing in upper layers.

The presentation layer follows a Single Page Application approach built with **Vite**, a frontend tool that starts a development server that processes files only when they are requested, rather than rebuilding the entire application after each change. Geospatial rendering uses **MapLibre GL**, while **deck.gl** aggregation layers support efficient heatmap visualisation at scale.

The anomaly detection service is implemented in **Python** with **Flask** and uses data and machine-learning libraries (`pandas`, `numpy`, `scikit-learn`). The model is loaded once at startup and reused for predictions, avoiding repeated file loading and reducing inference latency.

The following subsections detail the implementation of the common operation layers that manage this data flow, followed by an in-depth analysis of the implementation within each core Microservice.

5.2 Platform Common Operation Layers

The "front door" of the system is managed by two critical layers: the UI Layer (the user interface) and the API Gateway (the traffic controller). These layers work together to provide a seamless experience, abstracting the complexity of the distributed backend from the user.

5.2.1 The User Interface Layer Implementation

The **UI Layer Architecture**, detailed in Figure 17, is responsible for rendering the user interface. To ensure optimal user interaction, it is centred around a *Base Map* (powered by MapLibre GL) that visualises multiple complex data layers, including *Charger Markers* and a *Heatmap*. The *User Interaction Handler* is the logical core of the frontend, its *Event Capture* component intercepts user actions, which are then translated into structured API calls by the *Request Formatter* before being sent to the gateway. This separation of view and logic is fundamental to maintaining a responsive and maintainable user interface.

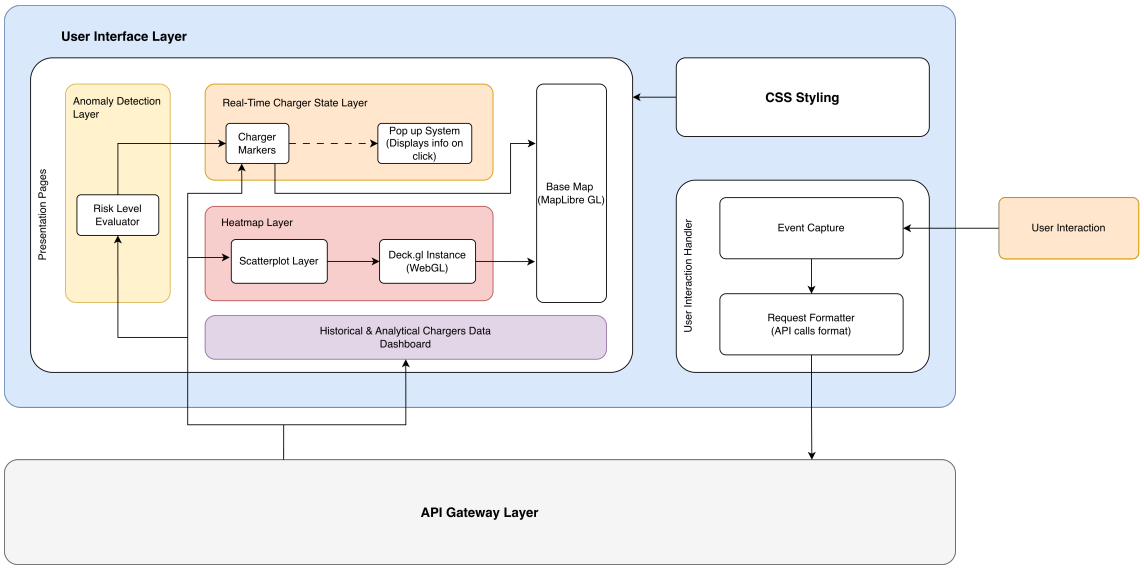


Figure. 17: UI Layer Architecture

The technical implementation of this layer utilises a Single Page Application (SPA) approach. The key challenge was to visualise distinct dimensions of data, including real-time status and historical patterns, simultaneously without compromising the browser. To address this rendering demand, the

implementation utilises **MapLibre GL**, a WebGL-based library that renders map elements using the computer's Graphics Processing Unit (GPU) rather than the Central Processing Unit (CPU). This allows the platform to visualise an extensive set of dynamic markers and complex heatmaps with smooth performance. The logic is structured around an *Event-Driven State Management* system, in which an event is fired when a user clicks a charger or selects a date range, triggering a specific API call. This ensures the interface remains "reactive," updating only the specific components that need to change rather than reloading the entire page.

5.2.2 The API Gateway Implementation

The **API Gateway Architecture**, shown in Figure 18, functions as the system's *central nervous system*. As illustrated in the diagram, a request enters the *Request Router*, which utilises an *Endpoint Mapper* to identify the target Microservice. The request is then passed to the *Request Dispatcher*, which constructs and forwards the final request. On the return path, the *Service Response Processing* block ensures all data is standardised before being sent back to the client.

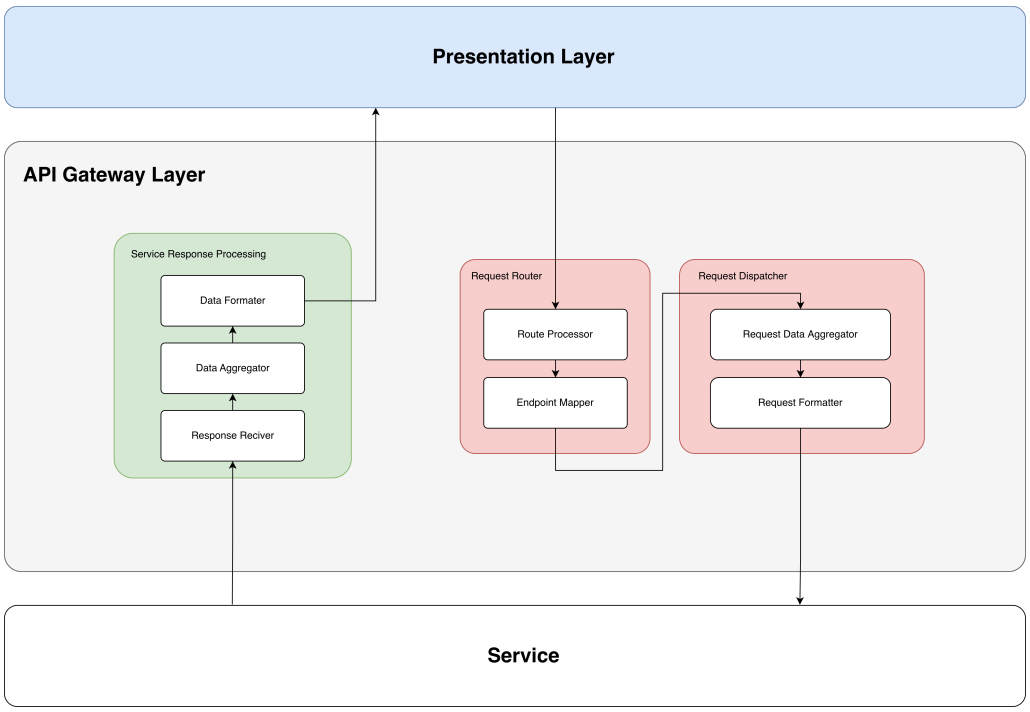


Figure. 18: API Gateway Layer / Middleware Architecture

Implemented in Node.js, the Gateway provides a single unified entry point for all client requests. Its internal logic functionality relies on **Request Piping**, a technique that streams data directly

from the client to the destination service without buffering it in memory. This ensures minimal latency and delay as the Gateway never holds the data.

Technically, this is achieved by establishing a direct read-write stream between the incoming client request and the backend service response (Appendix C.1 for the implementation details). This continuous stream centralises protocol adoption and error handling, ensuring that if a backend service fails, the frontend receives a standardised error message rather than a connection timeout, supporting the system's Reliability requirement.

5.3 Backend Services Layer

The intelligence of the platform resides in the Backend Services Layer. This section dives into the specific engineering challenges addressed within each functional domain, detailing how distinct Microservices were developed to solve problems.

5.3.1 Real-Time Charger's State Service

The **Real-Time Charger's State Service**, as represented by Figure 19, is designed for high performance and responsiveness using a **"Cache-Aside" pattern**. As outlined in the schematic diagram, the flow is coordinated by a *Cache Manager*. When a *Request Receiver* gets a request, it is passed to the *Cache Manager*, which checks the *Cache Data Store*. If the data is valid, it is served immediately. If the data is outdated, the manager dispatches a request call to the external *Mobi.e API*, more precisely, the Live Status Endpoint, which provides a snapshot of the network's state at the exact moment the request is made to update the cache. The *Data Processor* then consumes this data, using its *Markers Coordinates* and *Charger States* components to feed a *Data Normaliser*, which constructs the standardised information object for the UI.

The primary challenge for this service was balancing data synchronisation with API limitations. The external data provider, developed to access Mobi.e data, has rate limits, meaning the system cannot query it every time a user refreshes the page. To solve this, the service implements a **"Cache-Aside" Strategy**.

This design introduces a temporary memory store (cache) within the service. When a request arrives, the service logic evaluates the temporal validity of the stored record by comparing the current system time against the timestamp of the last successful fetch. If the cache duration has

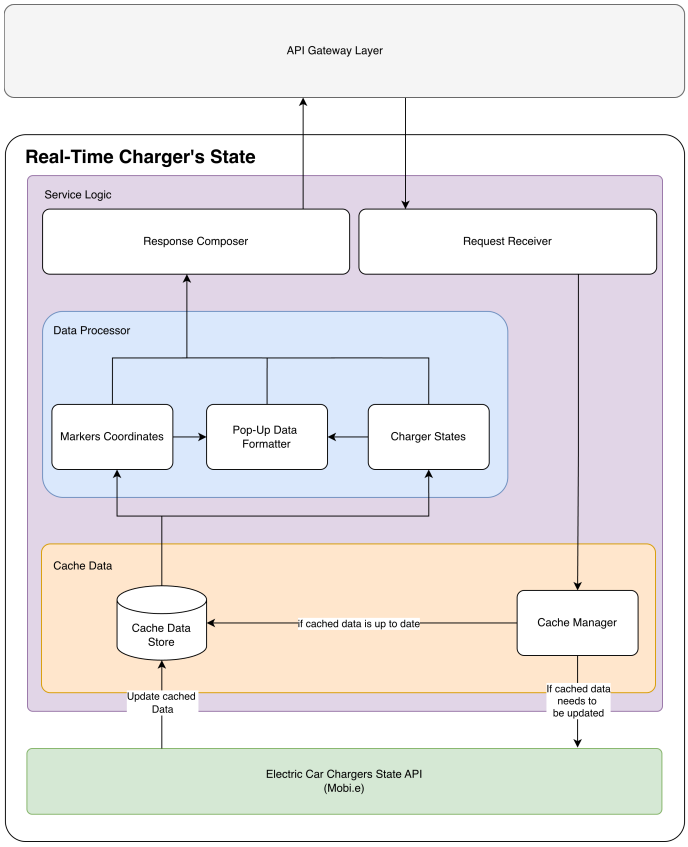


Figure. 19: Real-Time Charger's State Service Architecture

not surpassed its predefined valid window, the data is returned from memory, ensuring near-instantaneous latency. If, however, the cache duration outlasts the configured interval, the system treats the data as "obsolete" and executes a fetch operation to the external API provider, refreshing the cache. (Appendix C.2 for the specific implementation logic)

Beyond its caching mechanism, the service also performs **Data Normalisation**. The raw external location data often contains inconsistent status codes or multiple chargers with potentially conflicting states, represented in nested structures. This service parses and simplifies the received data structure into a standardised format before sending it to the frontend (see Appendix C.3 for the normalisation logic).

To address the referred nesting ambiguities, the service aggregates all chargers' inner states into a single, decisive *overallStatus* for the map marker. This aggregation follows a strict hierarchical logic: a "MIXED" status is assigned when conflicting states coexist (e.g., one available, one busy), followed by a priority check in which "Available" overrides "Charging" for uniform locations. This ensures the map marker always reflects the most relevant operational context for the user.

5.3.2 Database Sync Service (Data Ingestion)

Serving as the foundation for all historical analysis, the **Database Sync Service** is responsible for the systematic ingestion of data into the platform's long-term storage. As represented in Figure 20, it acts as an autonomous "Writer" service, populating the shared *PostgreSQL* database that is subsequently consumed by both the Data Analysis Dashboard and Heatmap services.

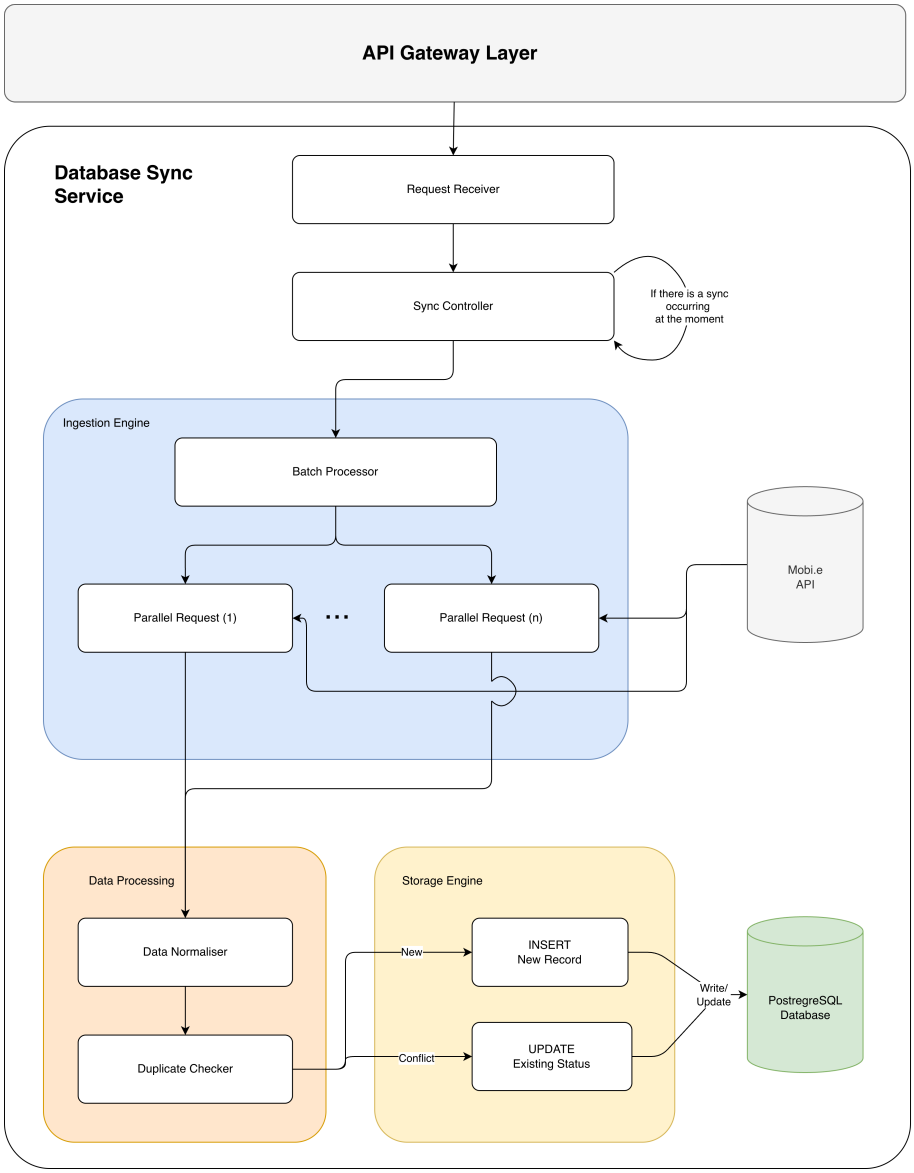


Figure. 20: Database Sync Service Architecture

The principal difficulty in this component lay in establishing a local, "*queryable*" dataset capable of supporting complex analytics using the chronological logs from the Mobi.e Historical Endpoint, which would be infeasible to perform by directly operating with the external API due to strict pag-

ination constraints (limited to 500-record batches) and the resulting latency of sequential fetching. To achieve this synchronisation efficiently, the implementation utilises a **Parallel Pagination Strategy**. The service's *Batch Processor* first queries the total number of records, then launches multiple *Parallel Requests* to fetch all data segments simultaneously. By leveraging asynchronous request handling patterns (detailed in Appendix C.4), the system processes these batches concurrently rather than sequentially. This reduces the total synchronisation time substantially compared to a linear fetching approach.

Furthermore, as shown in the diagram's *Storage Engine*, the data integrity logic relies on an **Conflict-Resilient Insertion** strategy. The system separates the flow into an 'INSERT' path for new records and an 'UPDATE' path for conflicting IDs. Using the SQL command `ON CONFLICT DO UPDATE`, the system ensures that if a record is fetched twice, it is simply updated rather than duplicated. This makes the system resilient to network retries and partial failures.

5.3.3 Data Analytics Dashboard Service

While the Sync Service handles writing, the **Data Analytics Dashboard Service** acts as a specialised "Data Accessor", serving as the dynamic query engine for the platform's statistical interface. As shown in Figure 21, the service is designed to translate high-level user requests, such as filtering by date range or specific charger IDs, into optimised database operations. This design exposes flexible API endpoints that allow the frontend to dynamically request precise subsets of data, abstracting the complexity of the underlying database schema.

Instead of fetching all raw data and filtering it in the application logic, the service's *SQL Query Builder* dynamically constructs optimised SQL queries. The system conditionally builds a query string by conditionally appending SQL 'WHERE' clauses based on the user's active filters, such as specific charger IDs or date ranges (see Appendix C.5 for the query construction logic). This method benefits from the database's native indexing capabilities, capable of filtering millions of rows in milliseconds, delivering fast analytics to the user.

5.3.4 Heatmap Service

Functioning as the platform's geospatial counterpart to the Data Analytics Dashboard, the **Heatmap Service** (detailed in Figure 22) is optimised for spatial rather than statistical querying. While the Dashboard Service focuses on temporal trends (e.g., "sessions per day"), this service utilises an

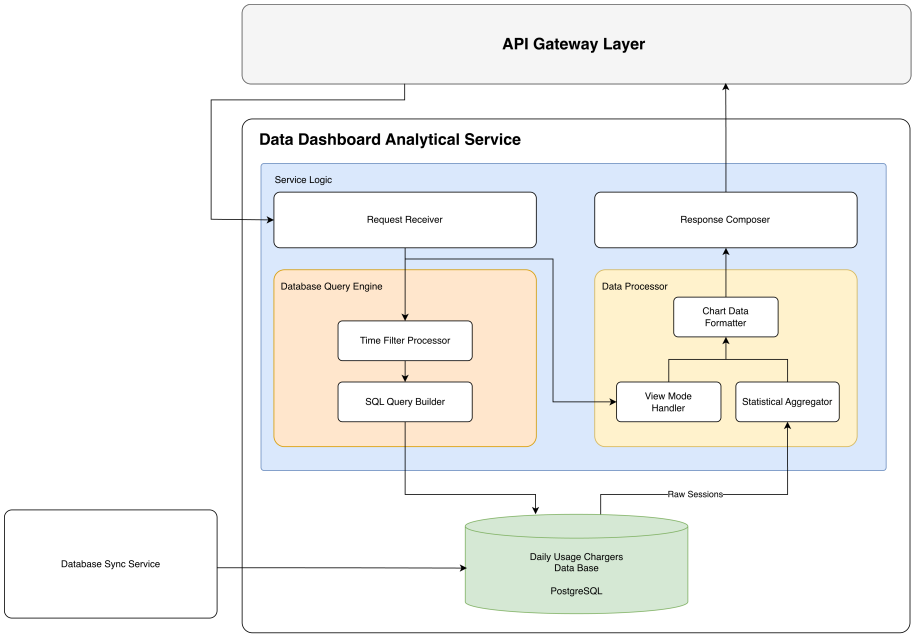


Figure. 21: Data Analytics Dashboard Service Architecture

In-Database Aggregation pattern to synthesise vast quantities of records into a coherent geographical intensity layer.

The major structural obstacle regarding this service was the extensive volume of data, as transferring raw session logs from every charging location to the client would require streaming substantial amounts of uncompressed data, inducing UI latency and compromising the application's responsiveness. In response to this, the implementation moves the computational load from the client to the database. The service's *Database Query Engine* executes an aggregation command that groups charging sessions by their geographic location (`GROUP BY location_id`) and sums their duration. This shifts the heavy lifting to the database engine, which returns only a summary list of coordinates.

Subsequently, the *Heatmap Layer Generator* then processes this raw aggregation into a standardised geospatial format optimised for WebGL. The service constructs a lightweight JSON data structure containing only the essential coordinates and intensity-weight values, discarding unnecessary data that could degrade performance (as detailed in Appendix C.6). This pre-formatting is essential for the frontend's Deck.gl engine, allowing it to instantly render complex intensity gradients without needing to perform CPU-intensive calculations on the browser side.

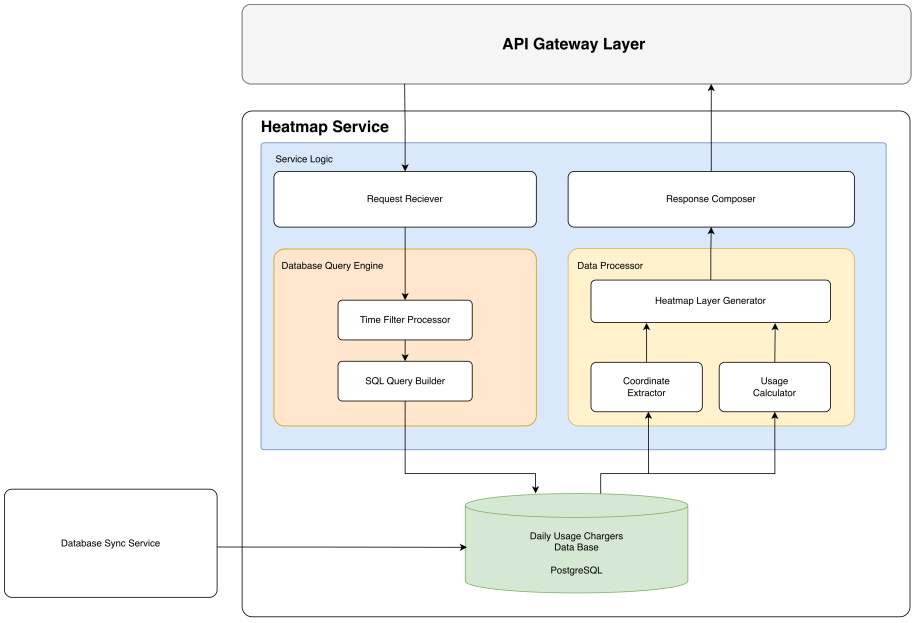


Figure. 22: Heatmap Service Architecture

5.3.5 Anomaly Detection Service

Finally, the **Anomaly Detection Service**, shown in Figure 23, introduces predictive analytics capabilities to the platform. While previous services rely on deterministic rule-based logic, this component integrates probabilistic ML models to identify possible irregularities in charging behaviour. The service is implemented in **Python** using the **Flask** framework. This technological foundation was chosen due to its native support for the project's ML libraries (such as *scikit-learn*) while exposing a standard HTTP interface, enabling a seamless integration with the platform's Node.js ecosystem.

To ensure real-time responsiveness, the service architecture separates model initialisation from request processing. Upon startup, the *Model Loader* initialises the system by deserialising the pre-trained model file (‘.pkl‘) directly into memory. This "Pre-loading initialisation" strategy means the resource-intensive process is done only once when the server boots, allowing subsequent user requests to be processed instantly.

When a prediction request arrives, the system follows a strict three-step pipeline. First, the *Feature Extractor* transforms the raw charging data into numerical vectors, ensuring the input format matches the training data exactly. Next, the *Model Predictor* runs these data points through the hybrid algorithm (combining K-Means and Isolation Forest). Finally, the *Risk Assessor* interprets

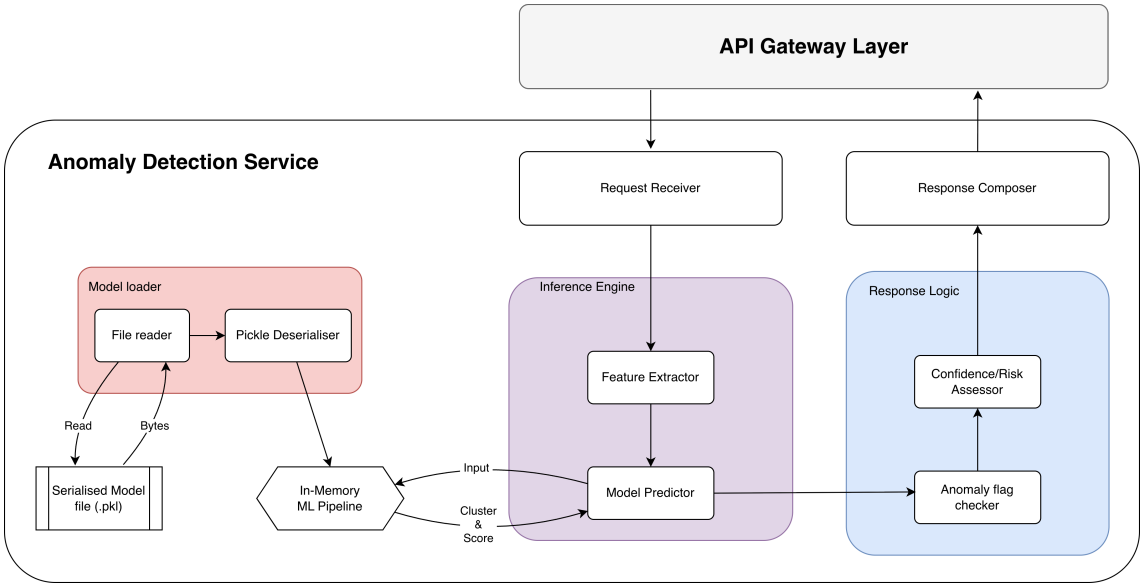


Figure. 23: Anomaly Detection Service Architecture

the model's mathematical output (see Appendix C.7 for the specific inference pipeline), classifying it as normal or suspicious and constructing the final JSON response for the frontend.

6 Evaluation Methodology

This chapter details the validation strategies set to evaluate the operational capability of both the analytical algorithms and the DS platform. To ensure a comprehensive assessment, the evaluation is structured into two distinct domains.

Section 6.1 establishes the statistical methodology used to validate the **Hybrid Anomaly Detection Algorithm**. Given the unsupervised nature of the problem and the lack of pre-labelled data, this section defines the construction of a deterministic Proxy "Ground Truth" baseline derived from physics-based rules. It further details the transformation of the dataset into a binary classification, which is necessary for calculating quantifiable efficacy measures such as **Precision**, **Recall**, and the **F1-Score**.

Subsequently, Section 6.2 shifts the evaluation to the DS Platform, specifically verifying the data transformation pipeline, designed to abstract the structural complexity of raw data. Given the system's current pre-production context, quantitative load testing is reserved for the future deployment phase. Instead, the validation focuses on the system's capacity to successfully represent charging records as standardised feature vectors, essential for the frontend application, as it enables clear, coherent visualisation of complex charging events, significantly simplifying data presentation.

6.1 Hybrid Anomaly Detection Algorithm

To formally measure the efficacy of the proposed Hybrid Unsupervised Learning approach, a quantitative validation framework was established. Since the dataset lacks pre-existing "ground truth" labels verified by capable operators (as is typical in unsupervised anomaly detection), the **Rule-Based Physics Analysis** (defined in Section 4.3) was utilised as the reference standard for a **Proxy Ground Truth**. Specifically, sessions identified with a high confidence of violation, defined as "Majority Abuse" or higher (flagged in at least 2 out of 3 power scenarios), were treated as confirmed positive cases of abuse $(y_{true} = 1)$.

The validation process followed a structured protocol to translate the qualitative outputs of the different models into comparable quantitative metrics.

6.1.1 Binary Classification Transformation

The first step was to coordinate the output formats of the two systems. While the models produce detailed categorical insights (e.g., "Cluster 2", "High Risk"), statistical validation requires mapping these states to a binary domain.

In this context, it is important to clarify the definition of the target classes. Following standard anomaly detection conventions, the Positive Class (1) does not imply a favourable outcome, but rather the presence of the anomaly (in this case, the event of interest). Consequently, the Negative Class (0) represents the inexistence of the anomaly (standard behaviour).

6.1.1.1 Reference Vector Construction (y_{true})

Derived from the strict Rule-Based Physics Analysis, this vector serves as the ground truth. Sessions classified as Majority Abuse (flagged in 2 or more scenarios) are assigned to Class 1 (Positive), indicating a confirmed abuse. All other sessions, including those with only minor or ambiguous flags, are assigned to Class 0 (Negative). This ensures that the baseline remains conservative, treating edge cases as normal to avoid artificially inflating the detection scores.

6.1.1.2 Prediction Vector Construction (y_{pred})

Derived from the Hybrid Anomaly Detection algorithm, this vector represents the model's output. Sessions where the Consensus Logic identified a possible threat (High or Medium Risk) are assigned to Class 1 (Positive), whereas sessions classified as Low Risk are assigned to Class 0 (Negative).

6.1.2 Evaluation Metrics

After converting the dataset into aligned binary arrays, performance was evaluated using the **Confusion Matrix** and calculated statistical metrics derived from its resultant quadrants.

The confusion matrix categorises the alignment between the Hybrid Model's predictions and the deterministic baseline. The structural layout is defined in Table 5:

The interpretation of each quadrant is contextualised within the specific operational constraints of parking management:

n = [Total Samples]		**Predicted Class (Hybrid Model)**	
		Normal (0)	**Abuse (1)**
Actual Class (Proxy Ground Truth)	**Normal (0)**	True Negative (TN)	False Positive (FP)
	Abuse (1)	False Negative (FN)	True Positive (TP)

Table 5: Confusion Matrix Structure: Hybrid Model vs. Rule-Based Baseline

- **True Positives (TP):** Sessions where the Hybrid Model successfully identified a violation that was confirmed by the baseline rules.
- **False Positives (FP):** Occur when the model flags a session as abusive, but the strict rules did not. In unsupervised learning, these are significant as they often can represent "implicit anomalies" or new behaviours that rigid rules fail to capture.
- **False Negatives (FN):** Indicate missed detections where the model failed to flag a confirmed abuse case.
- **True Negatives (TN):** Instances where both systems correctly agreed on the normality of the session.

From these components, three fundamental metrics are derived to assess the model's distinct capabilities.

6.1.2.1 Precision (Reliability)

This metric quantifies the credibility of the system's alerts. It answers the question: *When the system flags a session as abusive, what is the probability that this decision aligns with the established proxy ground truth?*

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

6.1.2.2 Recall (Sensitivity)

This metric assesses the coverage of the detection engine. It answers the question: *What proportion of the confirmed baseline violations did the model successfully identify?*

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

6.1.2.3 F1-Score

The F1-Score is the *"harmonic mean"* of Precision and Recall. It is the primary metric used for optimisation in this study because it penalises the model for ignoring anomalies (low recall) or generating excessive noise (low precision), providing a single scalar value of overall performance.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

6.1.3 Experimental Scenarios and Feature Availability

While the metrics defined above provide the mathematical tools for evaluation, to fully evaluate the impact of the "Energy Blindness" identified in the problem statement and measure the proposed solution's ability to overcome it, the experimental testing process was conducted across three distinct experimental configurations. These scenarios serve to benchmark the system's performance under different levels of data availability.

- **Scenario A: Time-Only Baseline:** This scenario simulates the current operational limitations of the public network. The models (K-Means and Isolation Forest) are trained and tested using exclusively the **Inference Vector (X_{inf})**, which contains only temporal features (Duration, Start Hour, Day of Week). This establishes the baseline performance under the current operational constraints.

- **Scenario B: Unconstrained Feature Set:** This scenario assumes a hypothetical environment with full smart-metering support. The models utilize the full **Calibration Vector (X_{cal})**, which includes energy consumption (kWh) and efficiency rates (η). This serves as a comparative benchmark to quantify the specific impact of missing telemetry, isolating the limitations imposed solely by hardware constraints.

- **Scenario C: Hybrid Operational Reality (Proposed Solution):** This scenario represents the specific contribution of this thesis. It evaluates the **Hybrid Strategy** in its intended deployment mode, utilising *Energy Data* during the pre-operational learning to calibrate the detection logic, but relying solely on *Time Data* for runtime inference. This assesses the system's ability to transfer knowledge from historical energy records to real-time temporal detection.

6.2 Digital Shadow Platform Methodology

Given the current pre-production status of the DS platform—operating in a local development environment rather than a distributed cloud infrastructure—standard non-functional testing metrics such as load balancing, latency analysis, and high-availability stress testing are reserved for the future deployment phase. Consequently, the evaluation methodology for the DS focuses strictly on **Functional Verification** and **Cognitive Load Reduction**.

The primary objective of this validation is to demonstrate the platform's capability to abstract the structural complexity of raw IoT streams into coherent, actionable visual representations. To quantify this capability, a **Data Traceability Analysis** framework was established. This method isolates specific, complex data scenarios and tracks their lifecycle from the raw input state to the final visual output.

To ensure a comprehensive evaluation of the system's operational value, this verification is structured around three distinct dimensions of data abstraction:

- **Spatial Abstraction (Real-time Map):** Verifying the transformation of disjointed JSON coordinate streams into a unified, status-aware geospatial interface.
- **Statistical Abstraction (Heatmap Service):** Verifying the system's ability to aggregate high-volume historical records into an immediate visual density surface, eliminating the need for manual tabular analysis.
- **Temporal Abstraction (Data Dashboard):** Verifying the reconstruction of fragmented, event-based logs into coherent, human-readable session objects.

For each dimension, the validation criteria are defined as **Information Fidelity** (ensuring no critical data is lost during normalisation) and **Interpretability** (ensuring the visual output facilitates faster decision-making than the raw logs).

7 Results and Discussion

This chapter presents the empirical results derived from the validation strategies established in Chapter 6, as it quantifies the efficacy of the proposed solution across two distinct dimensions, analytical precision and operational viability.

The analysis begins in Section 7.1 with a comparative benchmarking of the Hybrid Strategy against each of the independent unsupervised methods, validating the premise that, in this specific case, algorithmic coupling delivers superior detection capabilities. This is followed by a comprehensive analysis of the Hybrid Model's specific performance metrics against the physics-based baseline, interpreting the implications of its precision and recall scores in the context of urban parking monitoring.

Finally, Section 7.2 presents the functional verification of the DS platform, confirming the validity of the data transformation pipeline required to support the analytical processes established in the DS platform.

7.1 Evaluation of the Anomaly Detection Strategy

This section presents the quantitative validation of the proposed detection engine. The analysis is conducted in two stages: first, a comparative benchmarking to validate the architectural choice, followed by a detailed examination of the Hybrid Model's operational profile against the Rule-Based Proxy Ground Truth.

7.1.1 Experimental Setup and Dataset Characterisation

Before presenting the performance metrics, it is essential to characterise the dataset. The evaluation utilised a real-world dataset of 5,985 charging sessions from Funchal, collected between April and June 2025. To mitigate chronological bias, a non-sequential data split was applied. The unsupervised models were trained using 3,910 sessions extracted from April and June. Subsequently, the inference phase was executed exclusively on the 2,055 valid sessions recorded during May, following the exclusion of 20 invalid or duplicate entries from the original 2,075. This split ensured the algorithms were tested on entirely unseen, non-chronological operational data.

7.1.2 Comparative Analysis: Individual vs. Hybrid Architectures

To validate the hypothesis that a combined approach delivers superior detection capabilities compared to isolated models, the performance of the **Hybrid Strategy** was benchmarked against its individual base algorithms, **K-Means Clustering** and **Isolation Forest**, operating independently using only Inference Vector (X_{inf}), as this represents the only feature set available in a "real-world scenario," effectively establishing the comparison between the proposed solution (*Scenario C*) and the current operational limited baselines (*Scenario A*).

To further validate the design decisions and quantify the impact of the available data, this comparative evaluation is expanded to include the theoretical "Energy-Aware" environment (*Scenario B*). This multi-scenario testing allows the analysis to evaluate the proposed Hybrid Solution (*Scenario C*) not only against current possible conditions, but crucially against the system's potential capability.

Table 6 summarises the resulting performance metrics, offering a comprehensive view of the detection capabilities across all three experimental contexts.

Beyond the specific values, the aggregated data highlights two fundamental behavioural patterns. First, the Hybrid Consensus consistently establishes the upper standard of performance in every tested environment. Whether operating in the time-restricted (*Scenario A*) or in the feature-complete (*Scenario B*) context, the combined approach consistently outperforms the individual algorithms, confirming that the consensus logic successfully mitigates the specific weaknesses of each isolated model.

Second, the results reveal a distinct performance hierarchy governed by the energy-aware Scenario B. Most notably, the proposed solution (**Scenario C**) demonstrates a clear superiority over the standard time-based baseline (*Scenario A*). Despite both configurations operating without energy metrics for inference, the Hybrid architecture achieves a higher operational standard, validating the structural combination of profiling and isolation as a viable pathway to reduce the reliance on energy metrics.

7.1.2.1 Analysis of Model Synergy

Beyond the quantitative scores, a closer examination of the data reveals a distinct performance hierarchy that validates the necessity of the consensus approach.

Experimental Context	Model Architecture	Precision	Recall	F1-Score
Scenario A (Time-Only Baseline)	K-Means	0.658	0.412	0.507
	Isolation Forest	0.676	0.382	0.495
	Hybrid Consensus	0.663	0.439	0.528
Scenario B (Unconstrained Feature Set)	K-Means	0.782	0.581	0.667
	Isolation Forest	0.759	0.488	0.595
	Hybrid Consensus	0.778	0.632	0.697
Scenario C (Hybrid Operational Reality)	**Hybrid Consensus (Proposed)**	**0.703**	**0.476**	**0.568**

Table 6: Comparative Performance Metrics across different Experimental Scenarios and Detection Strategies

The standalone **Isolation Forest**, despite being a well-established standard in anomaly detection literature, demonstrated the weakest overall performance ($F_1 = 0.495$). This underperformance was primarily driven by its **low Recall of** 0.382, the lowest among all tested configurations. This result actually emphasises the algorithm's primary vulnerability, whenever operating without a scenario-based contamination rate, it lacks the contextual awareness required to effectively separate noise from true anomalies, resulting in a significant number of missed detections.

In contrast, the **K-Means Clustering** algorithm proved to be a more robust standalone baseline, achieving an F1-Score of 0.507. By strictly segmenting the data based on behavioural density, it improved upon the Isolation Forest's sensitivity, raising the Recall to 0.412. However, this gain came at a cost, as its **Precision dropped to** 0.658, lower than the uncalibrated Isolation Forest (0.676). This trade-off suggests that while K-Means is better at identifying potential outliers, the model's inflexibility regarding cluster boundaries introduces more false positives and still fails to capture the more subtle cases.

The **Hybrid Strategy** validates the assumption, suggesting that these models work best alongside one another. By using the "qualified" K-Means profiler to dynamically calibrate the Isolation Forest, the system successfully revealed the latent potential of the detection engine. Consequently, the calibrated Isolation Forest not only recovered from its initial underperformance but surpassed both base models. This synergy delivered significant gains across the board, most notably regarding **Recall**, the Hybrid model achieved a +15.5% **improvement over K-Means** and a substantial +24.6% **increase over the standalone Isolation Forest**. Simultaneously, it secured the highest **Precision** (0.703), representing a +7.0% **improvement compared to K-Means** and a +4.1%

gain over the uncalibrated Isolation Forest. This confirms that a statistically calibrated detector is definitely superior to both profiling and non-directed anomaly detection.

7.1.2.2 Impact of Energy Telemetry

Extending the analysis to the experimental scenarios, the superiority of the Hybrid Architecture remains evident regardless of feature availability. The trend observed in the time-only context repeats in the **Full-Feature** context, where the Hybrid approach achieves the highest overall score ($F1 = 0.697$).

Comparing the proposed solution (**Scenario C**) against a full-feature baseline (**Scenario B**) validates the specific impact of "Energy Blindness." As expected, the inclusion of energy telemetry produces superior performance metrics. However, the **Hybrid Operational Reality** model still maintains a competitive operational profile ($F1 = 0.568$), specifically preserving a high Precision (0.703). This indicates that while the lack of energy data results in a lower Recall (missing complex anomaly energetic patterns that require metering to detect), the Hybrid Consensus logic mitigates this deficit by enforcing high reliability for the detected anomaly events. Consequently, while the Energy-Enriched model represents the future theoretical maximum, the Time-Only Hybrid model is validated as a capable and safe solution for immediate, real-world deployment.

7.1.3 Detailed Analysis: Hybrid Anomaly Detection

Having established the Hybrid Model as the superior architecture, a deeper analysis of its performance metrics was conducted against the Physics-Based baseline to understand its operational behaviour.

7.1.3.1 Confusion Matrix Results

The confusion matrix reveals that the model adopts a **conservative detection strategy**, prioritising high-confidence threats over generalised detection. This is evidenced by the significant disparity between divergent categories, as evidenced by the system generating far fewer False Positives (119) than False Negatives (311). This distribution indicates that the algorithm implicitly favours reliability, preferring to classify ambiguous or uncertain sessions as “Normal” rather than risking a false accusation against a legitimate user. The specific breakdown of predicted results against the ground truth is detailed below in Table 7:

2055	Predicted Normal (0)	Predicted Abuse (1)
Rule-Based Normal (0)	1363	119
Rule-Based Abuse (1)	311	282

Table 7: Confusion Matrix Results: Hybrid Model vs. Rule-Based Baseline

7.1.3.2 Metric Interpretation

Based on the values extracted from the confusion matrix, the specific performance scores are calculated, also indicating a distinct operational profile characterised by high credibility and selective sensitivity.

Recall

With a score of 0.476 (47.6%), the model identifies approximately half of the physics-verified abuse cases. This indicates that the algorithm behaves as a "Conservative Evaluator". Without the rigid and absolute delineations of the rule-based system, the ML model filters out the "unclear or uncertain" cases, such as sessions that technically overstayed by small margins. While this results in a lower detection rate, it can suggest the model could be effectively ignoring borderline cases that could be not only difficult to enforce but arguably unnecessary to penalise given their minor impact.

Precision (Reliability)

The model achieved a Precision of 0.703 (70.3%), meaning that when it flags a session as abusive, it aligns with the proxy ground truth over 70% of the time. This is the system's strongest metric. In the context of parking management, a False Positive (wrongly accusing a legitimate user) carries significantly higher operational costs than a False Negative, as every unjustified accusation requires manual intervention and may involve users in the process. Surpassing the 70% reliability threshold ensures that automated alerts generated by the system have a high probability of validity, minimising the risk of unjustified enforcement actions

F1-Score (Operational Balance)

The resulting F1-Score of 0.568 represents a promising operational balance for an unsupervised system. Achieving a score above 0.50 demonstrates that the temporal features alone, when processed through the combined model, can account for a considerable portion of the changes in abusive behaviour even without direct use of energy features. Crucially, given the lack of real ground truth,

deviations from the static rule-based baselines should be interpreted with caution. While these discrepancies largely reflect classification errors inherent to unsupervised clustering, the inflexibility of the baseline implies that the metric may also penalise not necessarily *abuses*, but rather borderline cases that the rule-based system overlooks.

7.1.3.3 Summary of Hybrid Strategy Performance

From these results, it can be deduced that the Hybrid Model is a suitable candidate for an **Initial Operational Deployment**. Although the system was not explicitly calibrated for the purpose, it favours *Precision* over *Recall*, as the **observed performance profile** of the established model results in a methodology optimised for **Operational Trust**.

While it identifies approximately half of the theoretical abuses, it does so with considerable certainty. The algorithm's conservative nature lies precisely in this trade-off: it detects fewer cases overall, but ensures that the flagged sessions are highly accurate. This design directly addresses our vision on the municipality's operational reality, where an agent or police officer is mobilised to the location to verify the infraction. A highly sensitive system would overwhelm the available agents with false alarms or ambiguous cases. By acting as a reliable mechanism that isolates clear infractions, it allows the municipality to set regulatory measures for the most flagrant abuses. Consequently, it prevents the allocation of officers and avoids the risk of generating a high volume of legitimate appeals from users caused by false-positive penalties, ensuring a safe transition to autonomous supervision.

7.2 Digital Shadow Functional Verification

Following the verification of the fundamental principles of the Hybrid Model, the evaluation now ascends to the presentation layer. This section focuses on the DS platform, specifically verifying its capacity to handle and display data streams.

Following the methodology defined in Section 6.2, the functional verification of the DS was conducted by isolating three representative scenarios. This analysis demonstrates the platform's efficacy in transforming operational data into understandable insights across the spatial, statistical, and temporal domains.

7.2.1 Spatial Abstraction: Real-Time Network Status

The first verification focused on the **Real-time Map Service**, validating its capability to normalise inconsistent data structures and provide immediate situational awareness of the charging network.

7.2.1.1 Input State: Structural Inconsistency and Complexity

The raw input derived from Mobi.e API (represented by Figure 24) is constituted by a paginated list of JSON objects, whose structure presents two significant interpretation challenges. First, the data is deeply nested, requiring the operator to parse through `evses` and `connectors` arrays to find a simple status code. Second, and more critically, the API response often exhibits structural inconsistencies where key attributes (such as `party_id` or `max_power`) appear at different hierarchical levels depending on the hardware manufacturer or operator.

```
},
{
    "_id": "67e8000e2bdf8fa76e85e98a",
    "locationId": "FNC-00054",
    "__v": 0,
    "createdAt": "2025-03-29T14:13:34.111Z",
    "data": {
        "party_id": "ECI",
        "city": "Funchal",
        "address": "Universidade da Madeira, 9020-105 Funchal",
        "mobie_voltage_level": "BTE",
        "last_updated": "2026-02-07T11:55:08.502Z",
        "coordinates": {
            "latitude": "32.65939",
            "longitude": "-16.92562"
        },
        "evses": [
            {
                "uid": "FNC-00054-01",
                "evse_id": "PT*ECI*E*FNC*00054*01",
                "status": "CHARGING",
                "last_updated": "2026-02-07T11:54:02.552Z",
                "connectors": [
                    {
                        "standard": "IEC_62196_T2",
                        "format": "SOCKET",
                        "max_voltage": 400,
                        "max_amperage": 32,
                        "max_electric_power": 22000,
                        "last_updated": "2025-05-28T15:20:44.405Z",
                        "tariffs": [
                            {
                                "type": "REGULAR",
                                "elements": [
                                    {
                                        "price_components": [
                                            {
                                                "type": "FLAT",
                                                "price": 0.1572,
                                                "vat": 22,
                                                "step_size": 1
                                            }
                                        ]
                                    }
                                ],
                                "start_date_time": "2025-05-28T15:11:44.000Z"
                            }
                        ]
                    }
                ],
                "capabilities": [
                    "REMOTE_START_STOP_CAPABLE",
                    "RFID_READER",
                    "UNLOCK_CAPABLE"
                ]
            },
            {
```


(a) Raw API Response (Paginated List)

```
            {
                "uid": "FNC-00054-02",
                "evse_id": "PT*ECI*E*FNC*00054*02",
                "status": "AVAILABLE",
                "last_updated": "2026-02-07T11:54:56.323Z",
                "connectors": [
                    {
                        "standard": "IEC_62196_T2",
                        "format": "SOCKET",
                        "max_voltage": 400,
                        "max_amperage": 32,
                        "max_electric_power": 22000,
                        "last_updated": "2025-05-28T15:20:44.405Z",
                        "tariffs": [
                            {
                                "type": "REGULAR",
                                "elements": [
                                    {
                                        "price_components": [
                                            {
                                                "type": "FLAT",
                                                "price": 0.1572,
                                                "vat": 22,
                                                "step_size": 1
                                            }
                                        ]
                                    }
                                ],
                                "start_date_time": "2025-05-28T15:11:44.000Z"
                            }
                        ]
                    }
                ],
                "capabilities": [
                    "REMOTE_START_STOP_CAPABLE",
                    "RFID_READER",
                    "UNLOCK_CAPABLE"
                ]
            }
        ],
        "mobie_max_power_allowed": 0,
        "id": "FNC-00054"
    },
    "updatedAt": "2026-02-07T11:55:17.995Z"
},
```


(b) Nested Connector Hierarchy

Figure. 24: Input State Complexity: The raw API response requires manual parsing of nested hierarchies and inconsistent attribute locations to determine charger status.

In this raw format, an operator is less able to intuitively mentally map the coordinates to a specific urban context. Furthermore, determining the overall status of a station with multiple connectors requires aggregating conflicting states, as one connector is available and the other one is occupied, into a single operational decision.

7.2.1.2 Output State: Normalised Visual Markers

The DS platform successfully abstracts this complexity by rendering the normalised data as **Colour-Coded Geospatial Markers**. The system's presentation layer automatically aggregates the sub-states of all connectors to determine a dominant status colour. This transformation reduces the interpretation effort from reading and parsing hundreds of text lines to a single visual scan. The operator can instantly evaluate the network's density and availability in Funchal without interacting with the underlying data structure, validating the **Spatial Abstraction** capability.

As illustrated in Figure 25, the interface demonstrates three specific functional achievements:

- **Status Aggregation:** The system correctly identifies the "Mixed" state of station `FNC-00054`. Instead of showing two conflicting rows of text, it renders a single **Yellow Marker**, instantly communicating that the station is partially occupied but still usable.
- **Contextual Filtering:** The sidebar controls allow the operator to filter by "Municipality Chargers" or specific cities. This visually excludes non-pertinent information, reducing the attention demand from assessing 122 locations to only the subset relevant to the current operational task.
- **Contextual Labelling:** The "popup" card translates raw technical specifications into clear labels, turning raw metrics into valuable assets for planning and decision-making regarding the chargers.

7.2.2 Statistical Abstraction: Demand Density

The second verification targeted the **Heatmap Service**, evaluating its ability to reveal historical usage patterns by transforming raw database records into a geospatial density surface.

7.2.2.1 Raw Data Entropy

The initial state of this service consists of the raw `charging_sessions` database table. As shown in Figure 26, this dataset contains thousands of unconnected records, chronologically sorted and

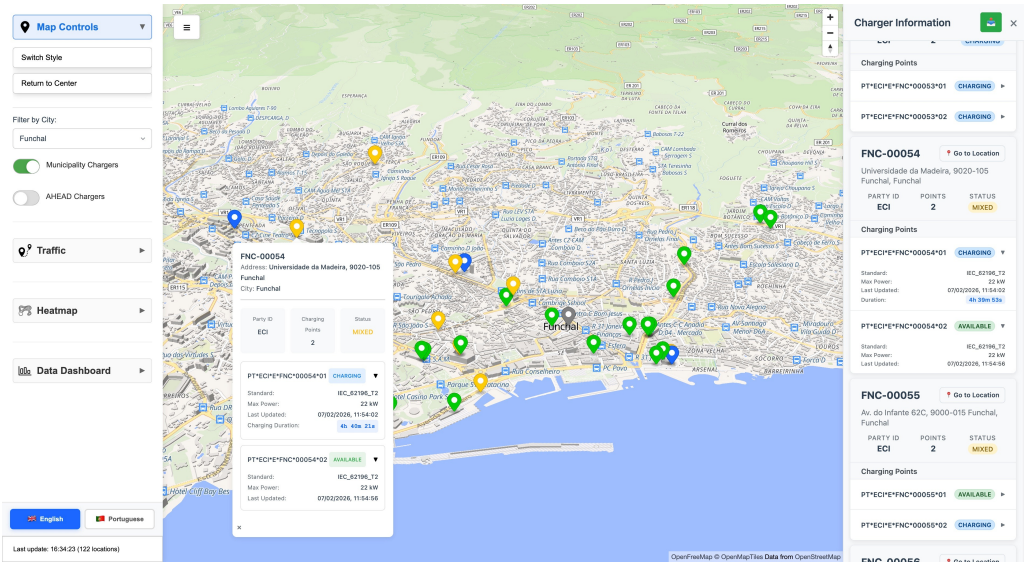


Figure. 25: Real-time Map Interface: Visualising the spatial abstraction of raw JSON data into aggregated status markers and filtered views.

distributed across hundreds of locations and evse IDs. In this format, identifying high-demand infrastructure is impractical, as the operator is presented with a continuous stream of start/stop timestamps without any spatial or cumulative context.

	session_id [PK] integer	location_id character varying (50)	evse_id character varying (50)	start_time timestamp without time zone	end_time timestamp without time zone	duration_minutes numeric (10,2)	terminated_by_user boolean	connector_standard character varying (50)
1	58522	FNC-00044	PT*FCT*E*FNC*00044*...	2026-01-01 00:42:36.234	2026-01-01 01:13:12.55	30.61	true	IEC_62196_T2_COMBO
2	55832	FNC-00047	PT*EMA*E*FNC*00047...	2026-01-01 01:11:16.116	2026-01-01 01:15:05.267	3.82	true	CHADEMO
3	53297	FNC-00049	PT*ECI*E*FNC*00049*01	2026-01-01 01:16:03.804	2026-01-01 04:09:07.755	173.07	true	IEC_62196_T2
4	54574	FNC-00057	PT*ECI*E*FNC*00057*01	2026-01-01 01:16:16.864	2026-01-01 03:57:40.397	161.39	true	IEC_62196_T2
5	54137	FNC-00053	PT*ECI*E*FNC*00053*02	2026-01-01 01:21:51.913	2026-01-01 02:15:19.374	53.46	true	IEC_62196_T2
6	54445	FNC-00055	PT*ECI*E*FNC*00055*02	2026-01-01 01:42:26.781	2026-01-01 04:20:21.863	157.92	true	IEC_62196_T2
7	57528	FNC-00070	PT*FAC*E*FNC*00070*...	2026-01-01 01:50:27.591	2026-01-01 13:41:34.669	711.12	true	IEC_62196_T2
8	53452	FNC-00049	PT*ECI*E*FNC*00049*02	2026-01-01 01:56:31.006	2026-01-01 02:04:17.731	7.78	true	IEC_62196_T2
9	54138	FNC-00053	PT*ECI*E*FNC*00053*02	2026-01-01 02:20:57.295	2026-01-01 09:34:32.213	433.58	true	IEC_62196_T2
10	53453	FNC-00049	PT*ECI*E*FNC*00049*02	2026-01-01 02:46:19.578	2026-01-01 11:21:50.692	515.52	true	IEC_62196_T2
11	58992	FUN-00023	PT*GLG*EFUN*00023*02	2026-01-01 03:58:29.472	2026-01-01 03:59:29.976	1.01	true	IEC_62196_T2_COMBO
12	55667	FNC-00040	PT*EMA*E*FNC*00040...	2026-01-01 04:03:56.261	2026-01-01 04:42:13.498	38.29	false	IEC_62196_T2_COMBO
13	53298	FNC-00049	PT*ECI*E*FNC*00049*01	2026-01-01 04:09:31.24	2026-01-01 08:01:03.83	231.54	true	IEC_62196_T2
14	54022	FNC-00053	PT*ECI*E*FNC*00053*01	2026-01-01 04:58:39.615	2026-01-01 09:34:44.234	276.08	true	IEC_62196_T2
15	53768	FNC-00052	PT*ECI*E*FNC*00052*01	2026-01-01 05:14:05.003	2026-01-01 15:43:03.627	628.98	true	IEC_62196_T2
16	55668	FNC-00040	PT*EMA*E*FNC*00040...	2026-01-01 07:36:59.576	2026-01-01 07:58:35.937	21.61	true	IEC_62196_T2_COMBO
17	54894	FUN-00020	PT*EDP*E*FUN*00020*...	2026-01-01 07:50:24.243	2026-01-01 15:39:14.343	468.84	true	IEC_62196_T2
18	53299	FNC-00049	PT*ECI*E*FNC*00049*01	2026-01-01 08:02:23.202	2026-01-01 13:30:55.305	328.54	true	IEC_62196_T2

Figure. 26: Raw Data Entropy: A sample of the disjointed charging session records before aggregation.

7.2.2.2 Input State: SQL Aggregation Queries

To extract meaning from this data volume, complex SQL aggregation logic is typically used. As illustrated in Figure 27, this involves a two-step task, first constructing the aggregation query (Figure 27a) to sum session durations by location, and second interpreting the resulting tabular output (Figure 27b).

While the resulting table represents a significant data reduction, condensing thousands of rows into a concise summary, it remains spatially abstract. A high usage value (e.g., 45,000 minutes) for `location_id="FUN-00025"` answers the question "How much?", but fails to answer "Where?", as it requires the operator to have memorised the physical coordinates of every charger in the network.

```sql
SELECT location_id,
       coordinates_lat,
       coordinates_lng,
       SUM(duration_minutes) as total_minutes
FROM charging_sessions
WHERE start_time BETWEEN '2026-01-01' AND '2026-01-31'
      AND (location_id LIKE 'FUN%' OR location_id LIKE 'FNC%')
GROUP BY location_id, coordinates_lat, coordinates_lng
ORDER BY location_id DESC
```

(a) Aggregation Logic (SQL)

	location_id character varying (50)	coordinates_lat numeric (10,6)	coordinates_lng numeric (10,6)	total_minutes numeric
1	FUN-00039	32.649690	-16.936080	45117.42
2	FUN-00038	32.651950	-16.926210	67561.00
3	FUN-00037	32.661251	-16.894082	2784.69
4	FUN-00036	32.660598	-16.893579	17092.91
5	FUN-00034	32.656198	-16.900474	222510.37
6	FUN-00033	32.649803	-16.903078	100158.72
7	FUN-00032	32.647499	-16.904372	96963.33
...				

(b) Tabular Output (PgAdmin)

Figure. 27: Intermediate Processing State: The transformation from raw logic to structured metrics. Despite the aggregation, the spatial context remains lost in the tabular format.

7.2.2.3 Output State: Density Surface

The platform completes the abstraction process by transforming this statistical table into a visual **Heatmap Layer**. By projecting the aggregated `total_minutes` onto the map surface as weighted intensity gradients, the system creates an immediate visual representation of demand.

As illustrated in Figure 28, this allows for instant identification of "Hot Zones" (areas of high utilisation in Red) and "Cold Zones" (underutilised infrastructure in yellow). This outcome confirms that the system successfully performs **Statistical Abstraction**, converting abstract numerical sums into actionable infrastructure planning insights that would be invisible in the intermediate tabular state.

7.2.3 Temporal Abstraction: Session Reconstruction

The final verification assessed the **Data Dashboard**, specifically its ability to reconstruct charging sessions from fragmented logs.

7.2.3.1 Reconstruction Pipeline: From Events to Sessions

The fundamental challenge with raw charging state transmission is that it is *event-based*, not *session-based*. As illustrated in the top row of Figure 29, the API returns a chronological stream of independent status updates. A single logical session is often distributed across dozens of log lines,

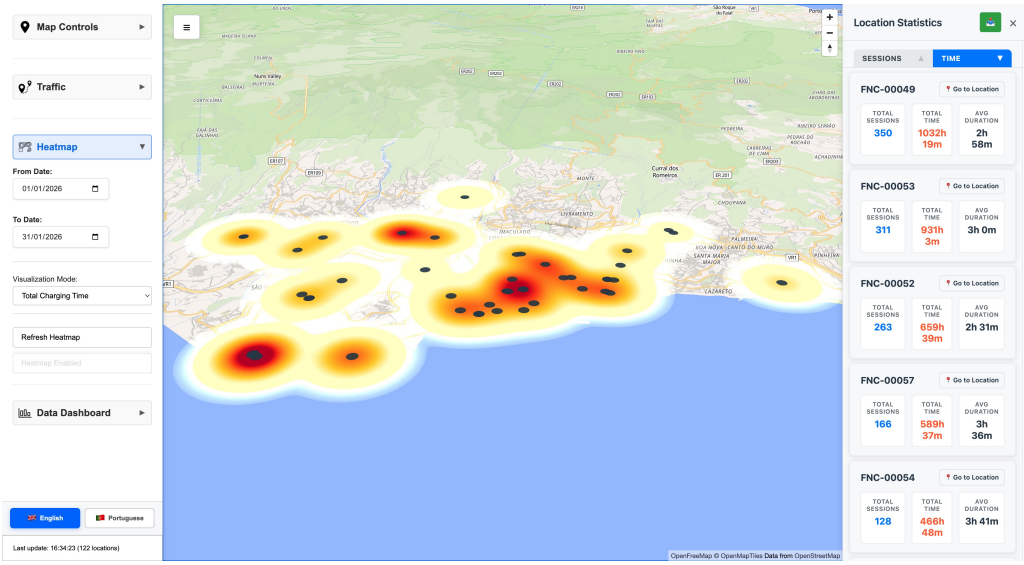


Figure. 28: Heatmap Service: Visualising the statistical abstraction. The abstract values from the SQL table are projected as a density surface, instantly revealing the high-demand corridor in the city centre.

requiring an operator to manually link a "Start" event (Figure 29a) with a "Stop" event (Figure 29b) while excluding intermediate states.

To address this, the platform implements a **Reconstruction Pipeline**. The ingestion engine analyses the event stream, identifies matching Start/Stop pairs for each specific EVSE ID, and merges them into a unified **Session Object**, which is then stored in the database (Figure 29c), the same database that serves as the starting point for section 7.2.2: Statistical Abstraction: Demand Density.

This process culminates in the **Interactive Data Table** (Figure 29d). Here, the operator no longer sees isolated "Events" but rather a logical "Session" with defined Start/End times and calculated durations. Crucially, this interface elevates the abstraction by providing temporal selection controls and multi-parameter filters, allowing the operator to isolate specific timeframes or chargers with minimal additional effort.

7.2.3.2 Multi-View Representation

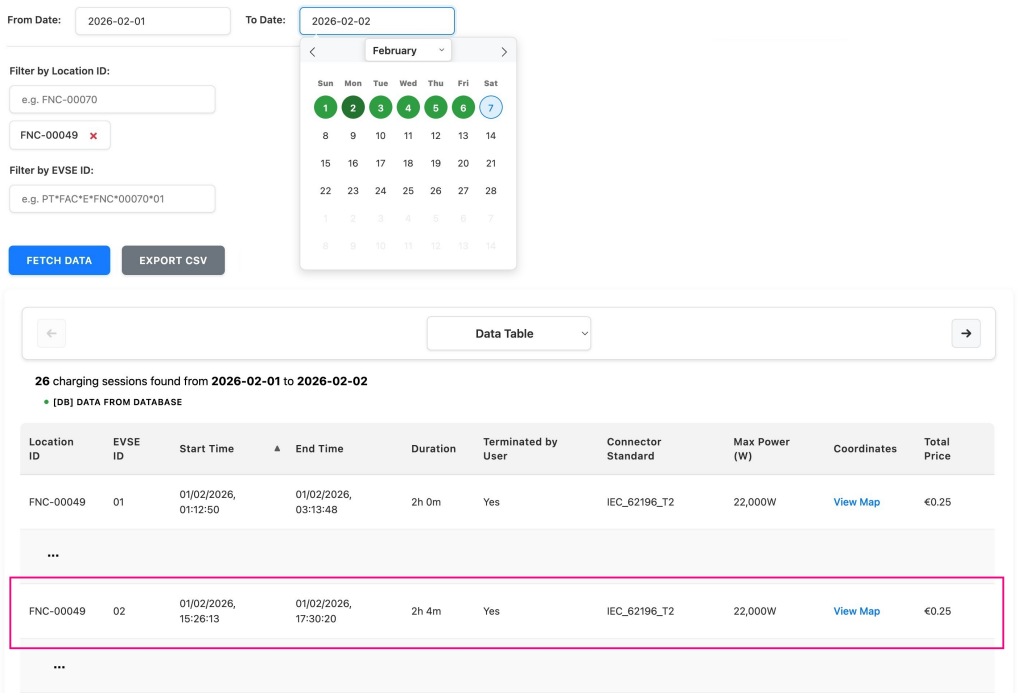
Once the data is abstracted into a coherent object, it becomes flexible enough to support multi-faceted data representations. As demonstrated in Figure 30, the platform dynamically transforms the tabular data into **multiple interactive formats simultaneously**. By plotting session data over time, this visual layer allows operators to instantly identify duration profiles (Figure 30a)

(a) Raw API: Start Event

(b) Raw API: End Event

	session_id [PK] integer	location_id character varying (50)	evse_id character varying (50)	start_time timestamp without time zone	end_time timestamp without time zone	duration_minutes numeric (10,2)	terminated_by_user boolean	connector_standard character varying (50)
94	67564	FNC-00040	PT*EMA*E*FNC*00040*02	2026-02-01 15:23:53.207	2026-02-01 15:30:16.814	6.39	false	IEC_62196_T2_COMBO
95	69663	FUN-00034	PT*GLG*EFUN*00034*01	2026-02-01 15:26:11.477	2026-02-02 03:05:25.577	699.24	true	CHADEMO
96	66054	FNC-00049	PT*ECI*E*FNC*00049*02	2026-02-01 15:26:13.082	2026-02-01 17:30:20.062	124.12	true	IEC_62196_T2
97	67429	FNC-00040	PT*EMA*E*FNC*00040*01	2026-02-01 15:31:29.138	2026-02-01 16:19:59.71	48.51	true	IEC_62196_T2_COMBO
98	67786	FNC-00048	PT*EMA*E*FNC*00048*01	2026-02-01 15:45:03.656	2026-02-01 16:28:11.474	43.13	true	IEC_62196_T2_COMBO
99	67565	FNC-00040	PT*EMA*E*FNC*00040*02	2026-02-01 15:47:53.881	2026-02-01 17:02:50.75	74.95	true	IEC_62196_T2_COMBO
100	67672	FNC-00047	PT*EMA*E*FNC*00047*01	2026-02-01 15:50:53.749	2026-02-01 15:51:16.637	0.38	true	CHADEMO
101	70579	FUN-00027	PT*MOT*E*FUN*00027*1	2026-02-01 15:52:52.527	2026-02-01 16:15:29.48	22.62	true	IEC_62196_T2
102	67673	FNC-00047	PT*EMA*E*FNC*00047*01	2026-02-01 15:53:55.933	2026-02-01 15:54:03.822	0.13	true	CHADEMO
103	68671	FNC-00061	PT*FAC*E*FNC*00061*01	2026-02-01 15:57:14.916	2026-02-01 17:41:18.822	104.07	true	IEC_62196_T2
104	67674	FNC-00047	PT*EMA*E*FNC*00047*01	2026-02-01 15:58:59.373	2026-02-01 15:59:37.686	0.64	true	CHADEMO

(c) Structured Database Table (PgAdmin)

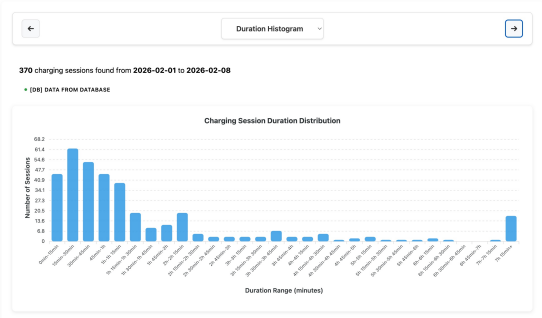


(d) Final Platform Interface with Filters

Figure. 29: Reconstruction Pipeline

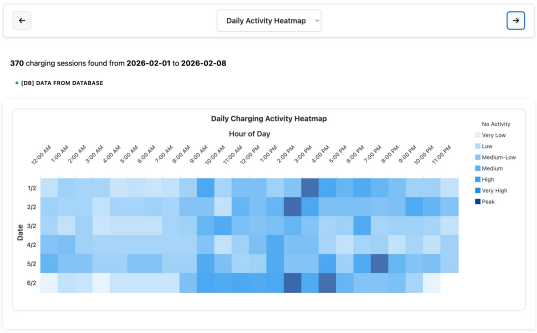
and daily peak hours, defined as the periods with the highest number of simultaneous sessions (Figure 30b and Figure 30c). While both scale the usage of chargers, they offer it different granularities: Figure 30b presents the average peak hour across the entire period, whereas Figure 30c details these peaks for each specific day and hour. These trends are further contextualised through location-specific volumes (Figure 30d) that would remain hidden in a standard spreadsheet.

This validates the **Temporal Abstraction** capability, proving that the platform not only reconstructs the past into actual charging sessions but formats it, facilitating the benchmarking of the chargers' network performance.

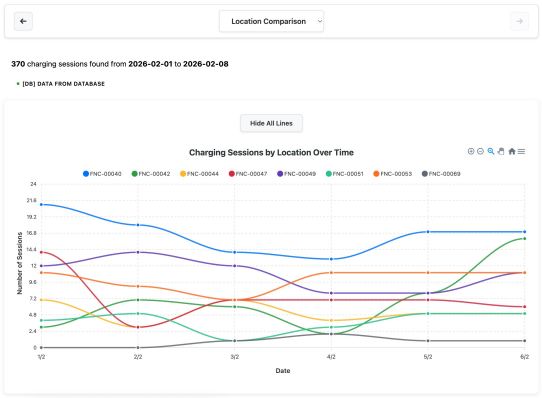


(a) Session Duration Distribution

(b) Daily Peak Hour Analysis



(c) Weekly Usage Heatmap



(d) Location-Specific Usage

Figure. 30: Multi-View Representation: The reconstructed session objects serve as the foundation for diverse analytical outputs, from visual trends to exportable datasets.

7.2.4 Verification of Information Abstraction & Decision Support Efficacy

The results obtained from these scenarios verify the DS's efficacy as the **Operational Interface** of the system. By successfully executing the Spatial, Statistical, and Temporal abstractions, the platform aligns the unprocessed equipment telemetry with valuable operational knowledge.

The verification confirms that the system effectively mitigates the analytical demand identified by the operators to manage raw data analysis. It functions as a unified operational environment, eliminating the previous requirement for the operator to navigate between distinct applications and API clients. It also redefines the operator's operations, transitioning from manual data parsing to a more efficient and autonomous practice. Consequently, the platform is validated as a **Decision Support Interface**, ensuring that the metrics generated by the core algorithms are exposed for immediate support in the municipality's decision-making.

7.3 Limitations of the Validation

Despite the successful outcomes demonstrating the system's potential, it is important to recognise the limitations of this study, which also present opportunities for future research.

1. **Scope of the Detection Model:** The Hybrid Anomaly Detection model, while statistically effective, relies on simplifications of the physical charging process. It does not yet account for all possible variables, most notably the impact of real-time energy fluctuations (amperage and voltage curves) on the definition of a "session", since no detailed energy telemetry was available from the source API. As a result, the model possesses an inherent "Energy Blindness," preventing the validation of specific abuse patterns, such as "Zombie" EVs that plug in without drawing power, that require direct metering to detect.

2. **Constraints of Pre-Deployment Status:** The validation environment, operating as a local instance rather than a distributed cloud infrastructure, imposed strict boundaries on the assessment of the **NFRs**.

   Consequently, attributes such as *Scalability* and *Performance* functioned primarily as architectural drivers that guided the system's foundational design decisions, rather than metrics that could be rigorously stress-tested. While these requirements were architecturally designed and theoretically validated, they remain untested under high concurrent user loads or long-term database growth. Furthermore, regarding *Usability*, although the initial interface concepts received preliminary validation from municipal stakeholders via mock-ups during an informal meeting, the final operational workflow remains a projection based on functional success, as it has not yet undergone formal testing in an active field deployment.

7.4 Chapter Summary

The empirical results presented in this chapter validate the two core components of the thesis.

First, the **Hybrid Anomaly Detection** strategy proved that algorithmic coupling outperforms standalone unsupervised methods. Additionally, by achieving a Precision of 70.3%, the model establishes a "High-Trust" baseline essential for automated supervision, effectively filtering operational noise while identifying actionable violations.

Second, the **Digital Shadow** verification confirmed that the functional abstractions successfully reduce the cognitive load of network management. The platform effectively translated the raw, fragmented complexity of the API into a coherent **Unified Operational Environment**.

In conclusion, these results confirm the project's success on both its analytical and operational representation fronts, demonstrating that the proposed solution satisfies the strict precision requirements for anomaly detection, despite the lack of energy data, as well as the usability requirements for a municipal charger's network monitoring tool.

8 Conclusion and Future Work

8.1 Summary of Work

This thesis presents the design, implementation, and validation of a **DS** platform for monitoring public EV charging infrastructure. Addressing the management challenges faced by the Municipality of Funchal, specifically the lack of real-time situational awareness and the inability to detect usage anomalies like "Idle Occupancy", the research delivered a dual-layer solution.

The work established a **Microservices-based Platform** to ingest and visualise disparate data streams, providing a unified "SSOT". Simultaneously, to address the "Energy Blindness" characteristic of public chargers, a **Hybrid Anomaly Detection Engine** was developed. By integrating "Energy-Aware" K-Means profiling when training with "exclusively temporal" Isolation Forest detection, the system successfully translated raw session records into actionable risk assessments.

8.2 Main Findings and Conclusions

The experimental validation supports three primary conclusions regarding the system's efficacy:

- **Superiority of the Hybrid Architecture:** The comparative analysis confirmed that the Hybrid Strategy ($F1 = 0.568$) outperforms standalone methods. By utilising pre-computed energy profiles to calibrate real-time temporal detection, the system achieved a **+15.5% improvement in Recall** over standard profiling and a **+24.6% increase** over the uncalibrated Isolation Forest.
- **Operationally Robust Configuration:** With a **Precision of** 70.3%, the detection engine reaches a satisfactory reliability level for automated supervision. The model effectively filters operational noise, ensuring that alerts sent to operators have a high probability of corresponding to genuine abuse.
- **Value of Digital Abstraction:** The functional verification of the DS proved that transforming fragmented API logs into **Geospatial Heatmaps** and **Session Objects** significantly reduces manual interpretation effort, validating the platform as an effective decision-support interface.

8.3 Limitations and Operational Constraints

Despite the successful validation, the development process exposed significant challenges that demonstrate the complexity of monitoring and assessing public infrastructure.

The primary constraint encountered was the **Inherent Energy Blindness**. As the source API does not provide real-time amperage or voltage telemetry, the detection model was forced to rely on "Probabilistic Profiling" (Time/Duration) rather than "Deterministic Measurement." While the Hybrid Strategy successfully mitigated this by inferring context from historical data, the system remains an *approximation* of the physical reality, unable to definitively confirm "Zero-Energy" events (Zombie EVs) in real-time.

Another significant limitation was the **Absence of an Empirical Ground Truth**. Due to the lack of official municipal records or verified labels identifying abusive charging sessions, the validation process relied on a "Rule-Based Physics Baseline" as the primary reference standard. While this provided a trustable representative, it remains a mathematical approximation rather than an absolute truth. This creates a situation where the developed Hybrid Model may identify subtle behavioural anomalies that the inflexible rule-based system fails to capture. In these instances, the Hybrid Model is effectively penalised with a lower statistical score for a "False Positive," even when it may have correctly detected a genuine abuse that the baseline was unable to recognise.

An additional limiting factor is the **Restricted Dataset Volume**. The anomaly detection models were trained using a dataset limited to exactly two months of operational logs. In the context of Unsupervised Machine Learning, this fundamental lack of data volume represents a significant constraint. Algorithms like K-Means rely on vast, diverse datasets to establish statistically robust clusters. Consequently, the models may overfit to the specific behaviours present during those specific two months, limiting the statistical confidence of the Isolation Forest and preventing the algorithms from accurately identifying broader operational trends.

An additional methodological limitation, this time regarding the ML algorithm, concerns to the **Linear Usage of Temporal Features**. In the current pipeline, cyclical temporal variables, such as the hour of the day or the day of the week, were preserved in their raw linear format (e.g., hours represented from 0 to 23). Consequently, distance-based algorithms like K-Means interpret the mathematical distance between 23:00 and 00:00 as 23 units, rather than the actual temporal distance of 1 unit. The absence of mathematical transformations prevents the model from correctly understanding the continuous, cyclical nature of time. This linear representation introduces artificial boundaries at midnight and at the end of the week, potentially distorting the cluster boundaries formed during the unsupervised learning phase.

Furthermore, the **Heterogeneity and Access Constraints** of the national network proved to be a substantial engineering challenge. The structural inconsistencies in the Mobi.e API required aggressive normalisation logic within the ingestion pipeline, specifically to resolve the ambiguous, nested charger states provided by the live snapshot endpoint. Concurrently, the historical endpoint's strict pagination constraints made direct analytical querying infeasible due to latency, forcing the architecture to separate data ingestion from analysis. This dual complexity confirms that creating a "DS" is not merely a data visualisation task but a complex data-related challenge, requiring continuous validation to prevent the impact of external schema changes from breaking the subsequent analytical models.

8.4 Future Research Directions

While the system establishes a robust foundation, the validation process identified specific areas for evolution.

8.4.1 Deploy the Application

The current validation was conducted in a local development environment, which imposes constraints on long-term data retention and concurrent user access. The next logical step is a full production deployment using a container-based deployment designed for the architecture. Migrating the Microservices to a distributed cloud environment would allow the platform to scale dynamically, fully validating the non-functional requirement of Scalability in a high-concurrency production setting.

8.4.2 Improve the Hybrid Model with energy Telemetry

As evidenced by the experimental benchmarks, the most significant jump in performance would come from bridging the "Energy Blindness". Since Section 7.1.2.2 have already proven that the method improves with access to an unconstrained feature set, if real-time telemetry becomes available via Smart Meter integration, the model should be updated to ingest amperage and voltage curves directly. This would allow the system to identify complex abuse patterns, such as EVs that connect without drawing power, potentially elevating the recall metric to much better levels.

8.4.3 Expand Dataset Volume for Pattern Discovery

Future research will address the current data limitations by integrating a recently acquired 37-month dataset. Although its late arrival prevented its inclusion in the present work, utilising these longitudinal records will allow the system to move beyond a static baseline to actively learn complex, long-term usage patterns. Specifically, an extended historical analysis will facilitate the identification of long-term seasonal fluctuations, distinct temporal dynamics (such as the contrast between daytime peaks and overnight parking), and critical spatial disparities (differentiating user behaviour in high-density urban centres if compared to isolated chargers). By determining how baseline occupancy patterns differ across these different times and locations, it is expected the anomaly detection thresholds can be adjusted to fit each specific context, significantly increasing the system's overall precision.

8.4.4 Implement Cyclical Feature Encoding

To address the current distortion in how the models evaluate temporal data, future iterations of the data pipeline must implement cyclical feature encoding. By applying sine and cosine mathematical transformations to temporal variables (such as the hour of the day and day of the week), the linear timeline will be mapped onto a continuous circle. This refinement will ensure that distance-based models like K-Means correctly interpret 23:00 and 00:00 as neighbouring data points. Implementing this transformation is a straightforward yet impactful algorithmic adjustment that will significantly improve the accuracy and cohesion of the behavioural clusters.

8.4.5 Contextualise & Refine the Proxy Ground Truth

Although the current rule-based proxy establishes a baseline, it lacks the contextual awareness. To bridge the gap between the current proxy ground truth and reality, future iterations must incorporate contextual urban information and institutional guidelines. Integrating the physical distance between nearest charging stations would contextualise occupancy durations by considering the lack of nearby charging alternatives, while also highlighting high-demand zones to guide the placement of new infrastructure. Additionally, dynamic activity metrics like Google Places "Popular Times" could provide the socio-urban context, allowing the system to correlate charging demand with the surrounding socio-economic activity. The model then, must be aligned with the specific regulatory definitions of "abusive occupancy" enforced established by the CMF.

8.4.6 Validate and Customise the Frontend

While the current dashboard (Figure 30) provides a comprehensive engineering view of the network, it may not perfectly align with the daily workflows of municipal enforcement teams. Future research should involve formal testing with municipality staff to develop **municipality-specific data representations**. As a concrete example of this necessary adaptation, the project's development highlighted the limitations of purely real-time anomaly alerts for physical policing. Consequently, a proposed improvement is the addition of an "Abuse Heatmap" to visually pinpoint the most frequently abused chargers. For instance, developing a simplified "action-oriented dashboard view" would complement the analytical views and enhance the platform's operational utility.

8.5 Closing Remarks

This thesis demonstrates that intelligent monitoring of public infrastructure does not require retrofitting the existing charging network or deploying additional hardware. By applying robust data processing and hybrid unsupervised learning techniques, it is possible to extract high-value operational insights from existing, limited data streams. The **DS** developed in this work serves as a proof-of-concept for an optimised, telemetry-based approach to urban management in Funchal, laying the groundwork for a more efficient and fairly distributed electric mobility network.

References

[1] R. Ruggieri, M. Ruggeri, G. Vinci, and S. Poponi, "Electric mobility in a smart city: European overview," *Energies*, vol. 14, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/2/315

[2] A. M. B. Francisco, J. Monteiro, and P. J. S. Cardoso, "A digital twin of charging stations for fleets of electric vehicles," *IEEE Access*, vol. 11, pp. 125 664–125 683, 2023.

[3] Edward H. Glaessgen and D. S. David Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," Apr. 2012, mAG ID: 2046074714.

[4] Michael W. Grieves and John Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," pp. 85–113, Jan. 2017, mAG ID: 2518003110.

[5] E. VanDerHorn and S. Mahadevan, "Digital twin: Generalization, characterization and implementation," *Decision Support Systems*, vol. 145, p. 113524, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923621000348

[6] M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi, "A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future," *IEEE Access*, vol. 11, pp. 17 471–17 484, 2023.

[7] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, and D. U. Sauer, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *Journal of Energy Storage*, vol. 30, p. 101557, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X20308495

[8] A. Julin, K. Jaalama, J.-P. Virtanen, M. Pouke, J. Ylipulli, M. Vaaja, J. Hyyppä, and H. Hyyppä, "Characterizing 3d city modeling projects: Towards a harmonized interoperable system," *ISPRS International Journal of Geo-Information*, vol. 7, p. 55, 02 2018.

[9] S. M. E. Sepasgozar, "Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment," *Buildings*, vol. 11, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2075-5309/11/4/151

[10] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, no. 1, p. 13, 2020. [Online]. Available: https://doi.org/10.1186/s40323-020-00147-4

[11] G. Schroeder, C. Steinmetz, C. E. Pereira, I. Muller, N. Garcia, D. Espindola, and R. Rodrigues, "Visualising the digital twin using web services and augmented reality," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 522–527.

[12] X. Ye, S. Jamonnak, S. Van Zandt, G. Newman, and P. Suermann, "Developing campus digital twin using interactive visual analytics approach," *Frontiers of Urban and Rural Planning*, vol. 2, no. 1, p. 9, 2024. [Online]. Available: https://link.springer.com/10.1007/s44243-024-00033-2

[13] J. L. D. Comba, N. O. Santos, J. C. Rivera, R. K. Romeu, and M. Abel, "Data visualization for digital twins," *Computing in Science Engineering*, vol. 25, no. 2, pp. 58–63, 2023.

[14] Y. Liu, X. Tu, D. Chen, K. Han, O. Altintas, H. Wang, and J. Xie, "Visualization of mobility digital twin: Framework design, case study, and future challenges," in *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 2023, pp. 170–177.

[15] R. Carley, S. Fuller, W. Bond, P. Jones, D. Allen, A. Jordan, and T. Falls, "Data analytics and visualization application for asset health monitoring," *Annual Conference of the PHM Society*, vol. 14, no. 1, 2022. [Online]. Available: https://papers.phmsociety.org/index.php/phmconf/article/view/3214

[16] S. Thelen, F. Eder, M. Melzer, D. Weber Nunes, M. Stadler, C. Rechenauer, M. Obergrießer, R. Jubeh, K. Volbert, and J. Dünnweber, "A slim digital twin for a smart city and its residents," in *Proceedings of the 12th International Symposium on Information and Communication Technology*, ser. SOICT '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 8–15. [Online]. Available: https://doi.org/10.1145/3628797.3628936

[17] V. Ramani, M. Ignatius, J. Lim, F. Biljecki, and C. Miller, "A dynamic urban digital twin integrating longitudinal thermal imagery for microclimate studies," in *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 421–428. [Online]. Available: https://doi.org/10.1145/3600100.3626345

[18] S. Ivanov, K. Nikolskaya, G. Radchenko, L. Sokolinsky, and M. Zymbler, "Digital twin of city: Concept overview," in *2020 Global Smart Industry Conference (GloSIC)*, 2020, pp. 178–186.

[19] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[20] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, "Microservices: yesterday, today, and tomorrow," 2017. [Online]. Available: https://arxiv.org/abs/1606.04036

[21] M. Parvizi, M. Noei, and M. Yalpanian, "Highly scalable smart metering iot platform based on microservice architecture," in *2022 Sixth International Conference on Smart Cities, Internet of Things and Applications (SCIoT)*, 2022, pp. 1–6.

[22] J. Lewis and M. Fowler, "Microservices: A definition of this new architectural term," *MartinFowler.com*, March 2014. [Online]. Available: https://martinfowler.com/articles/microservices.html

[23] B. Butzin, F. Golatowski, and D. Timmermann, "Microservices approach for the internet of things," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–6.

[24] C. Pahl, "Containerization and the paas cloud," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 24–31, 2015.

[25] D. Nicolas, R. De Jesus Marques, and C. Feltus, "A microservices approach to scenario-based integration of smart mobility digital twins," 09 2025.

[26] J. T. Zhao, S. Y. Jing, and L. Z. Jiang, "Management of api gateway based on micro-service architecture," *Journal of Physics: Conference Series*, vol. 1087, no. 3, p. 032032, sep 2018. [Online]. Available: https://doi.org/10.1088/1742-6596/1087/3/032032

[27] W. A. Ali, M. P. Fanti, M. Roccotelli, and L. Ranieri, "A review of digital twin technology for electric and autonomous vehicles," *Applied Sciences*, vol. 13, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/10/5871

[28] M. Dijk, R. J. Orsato, and R. Kemp, "The emergence of an electric mobility trajectory," *Energy Policy*, vol. 52, pp. 135–145, 2013, special Section: Transition Pathways to a Low Carbon Economy. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421512003242

[29] A. R. Mesquita, V. H. S. de Abreu, C. N. Poyares, and A. S. Santos, "Barriers to electric vehicle adoption: A framework to accelerate the transition to sustainable mobility," *Sustainability*, vol. 17, no. 18, 2025. [Online]. Available: https://www.mdpi.com/2071-1050/17/18/8318

[30] A. Tayri and X. Ma, "Grid impacts of electric vehicle charging: A review of challenges and mitigation strategies," *Energies*, vol. 18, no. 14, 2025. [Online]. Available: https://www.mdpi.com/1996-1073/18/14/3807

[31] C. Li, J. Lei, L. Yang, W. Xu, and Y. You, "Research on electric vehicle powertrain systems based on digital twin technology," *Electronics*, vol. 13, no. 20, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/20/4103

[32] Open Charge Alliance, *Open Charge Point Protocol (OCPP) Specification*, Open Charge Alliance, 2024, accessed: 2026-02-12. [Online]. Available: https://openchargealliance.org/my-oca/ocpp/

[33] M. Fischer, C. Hardt, W. Michalk, and K. Bogenberger, "Charging or idling: Method for quantifying the charging and the idle time of public charging stations," in *Proceedings of the TRB 101st Annual Meeting Compendium of Papers; Transportation Research Board: Washington, DC, USA*, 2022, p. 22.

[34] A. Lucas, R. Barranco, and N. Refa, "Ev idle time estimation on charging infrastructure, comparing supervised machine learning regressions," *Energies*, vol. 12, no. 2, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/2/269

[35] A. Biswas, R. Gopalakrishnan, and P. Dutta, "Managing overstaying electric vehicles in park-and-charge facilities," 2016. [Online]. Available: https://arxiv.org/abs/1604.05471

[36] D. Fährmann, L. Martín, L. Sánchez, and N. Damer, "Anomaly detection in smart environments: A comprehensive survey," *IEEE Access*, vol. 12, pp. 64 006–64 049, 2024.

[37] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. Salvo Rossi, "Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions," *Sensors*, vol. 23, no. 5, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2844

[38] S. M. Miraftabzadeh, C. G. Colombo, M. Longo, and F. Foiadelli, "K-means and alternative clustering methods in modern power systems," *IEEE Access*, vol. 11, pp. 119 596–119 633, 2023.

[39] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80 716–80 727, 2020.

[40] J. R. Helmus, M. H. Lees, and R. van den Hoed, "A data driven typology of electric vehicle user types and charging sessions," *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102637, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19315414

[41] M. Cederle, A. Mazzucco, A. Demartini, E. Mazza, E. Suriani, F. Vitti, and G. A. Susto, "Explainable anomaly detection for electric vehicles charging stations**this work was partially carried out within the italian national center for sustainable mobility (most) and received funding from nextgenerationeu (italian nrrp – cn00000023 - d.d. 1033 17/06/2022 - cup c93c22002750006)." *IFAC-PapersOnLine*, vol. 59, no. 26, pp. 301–304, 2025, 7th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896325027272

[42] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, Mar. 2012. [Online]. Available: https://doi.org/10.1145/2133360.2133363

[43] Z. Lin, X. Liu, and M. Collu, "Wind power prediction based on high-frequency scada data along with isolation forest and deep learning neural networks," *International Journal of Electrical Power Energy Systems*, vol. 118, p. 105835, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061519332491

[44] G. D. Gasperis and S. D. Facchini, "A comparative study of rule-based and

data-driven approaches in industrial monitoring," 2025. [Online]. Available: https://arxiv.org/abs/2509.15848

[45] Automóvel Club de Portugal, "Mobilidade elétrica em portugal em 2025," Observatório ACP, February 2025, accessed: 2026-02-12. [Online]. Available: https://www.acp.pt/o-clube/institucional/observatorio-acp/mobilidade-eletrica-em-portugal-em-2025

A Functional Requirements Specifications

This appendix provides the formal specification of the system's Functional Requirements (FR), derived from the collaborative validation process with the Municipality of Funchal. The requirements are categorised according to the core system capabilities.

A.1 Real-Time EVSE Infrastructure Monitoring

- **FR01:** The system shall display a geographically referenced, interactive base map of all public EVSE assets within the municipality.
- **FR02:** The system shall visually indicate the real-time operational status (e.g., Available, Charging, Out-of-Order) of each charging station using colour-coded markers.
- **FR03:** The system shall automatically aggregate conflicting connector statuses (e.g., one connector available, one charging) into a unified visual representation for the station marker.
- **FR04:** The system shall allow the operator to select a specific charging station to view detailed telemetry, including ongoing session durations and connector specifications.

A.2 Data Analytics Dashboard

- **FR05:** The system shall allow the user to filter historical charging records by defining specific date ranges.
- **FR06:** The system shall allow the user to filter historical charging records by specific EVSE identifiers or locations.
- **FR07:** The system shall dynamically generate visual charts (e.g., session duration distributions, daily peak hours) based on the active filters.
- **FR08:** The system shall allow the operator to export the filtered dataset into a standard, machine-readable format (e.g., CSV) for external analysis.

A.3 Geospatial Heatmap Visualisation

- **FR09:** The system shall generate a geospatial heatmap layer that can be toggled on or off over the primary map interface.

– **FR10:** The system shall calculate the heatmap intensity gradients based on the accumulated session duration (usage density) of the chargers within the user-defined temporal filters.

A.4 Anomaly Detection and Profiling

– **FR11:** The system shall assign a multi-level risk classification (e.g., High, Medium, Low) to evaluated sessions based on the consensus between the clustering and outlier detection models.

– **FR12:** The system shall visually flag locations or sessions identified as high-risk anomalies on the user interface.

B Anomaly Detection Algorithmic Implementation

This section details the algorithmic core of the Data Profiling strategy discussed in Section **??**. These listings demonstrate the implementation of the Hybrid Unsupervised Learning approach.

B.1 K-Means: Silhouette Optimization

```
def find_optimal_clusters(self, max_clusters=10):
    """
    Iterates through possible k values to find the best separation.
    """
    silhouette_scores = []
    # Test k from 2 up to max_clusters
    k_range = range(2, min(max_clusters + 1, len(self.scaled_data)))

    for k in k_range:
        # Initialise temporary model
        kmeans = KMeans(n_clusters=k, random_state=42)
        kmeans.fit(self.scaled_data)

        # Calculate Silhouette Score (s)
        # Higher score = better defined clusters
        score = silhouette_score(self.scaled_data, kmeans.labels_)
        silhouette_scores.append(score)

    # Select k corresponding to the maximum score
    best_k = k_range[np.argmax(silhouette_scores)]
    return best_k
```

Listing 1: Optimization Loop for Finding Best k

B.2 K-Means: Model Fitting

```
def perform_clustering(self, n_clusters):
    """
    Fits the final model using the optimised k
    """
    # Initialize the model with the best k found
    self.model = KMeans(n_clusters=n_clusters, random_state=42)

    # Fit the model to the data and assign labels
    # This minimizes the WCSS (J) for the chosen k
    self.clusters = self.model.fit_predict(self.scaled_data)

    return self.model
```

Listing 2: Final K-Means Model Fitting

B.3 Isolation Forest: Initialization

```
def fit_isolation_forest(self):
    # Initialize the ensemble with domain-specific constraints
    self.model = IsolationForest(
        contamination=self.contamination,  # Dynamic constraint
        n_estimators=100,                  # Stability via ensemble
        random_state=42
    )

    # Fit the trees to the dataset
    self.model.fit(self.scaled_data)
```

Listing 3: Isolation Forest Initialization

B.4 Isolation Forest: Scoring Logic

```
    # Predict anomalies: -1 for Anomaly, 1 for Normal
    # This applies the binary cut-off based on contamination
    self.predictions = self.model.predict(self.scaled_data)

    # Calculate raw decision scores
    self.anomaly_scores = self.model.decision_function(self.scaled_data)

    return self.predictions, self.anomaly_scores
```

Listing 4: Generating Anomaly Scores

B.5 Hybrid Strategy: Dynamic Calibration

```
def calculate_contamination_from_kmeans(self):
    """
    Derives contamination rate from K-Means cluster sizes.
    """
    # 1. Identify the 'Suspicious' Cluster
    # Defined as the cluster with the longest average session duration
    suspicious_cluster = max(
        cluster_stats.keys(),
        key=lambda x: cluster_stats[x]['avg_duration']
    )

    # 2. Calculate Base Rate (P(C_suspicious))
    # Percentage of total sessions falling into this suspicious cluster
    raw_abuse_rate = cluster_stats[suspicious_cluster]['percentage'] / 100

    # 3. Apply Confidence Adjustment (alpha = 0.85)
    # Reduces false positives by assuming not ALL sessions in the cluster
    are abusive
    adjusted_contamination = raw_abuse_rate * self.confidence_adjustment

    # 4. Apply Safety Bounds (clip between 5% and 25%)
    # Prevents model from being too lax or too aggressive
    final_contamination = np.clip(adjusted_contamination, 0.05, 0.25)

    return final_contamination
```

Listing 5: Hybrid Calibration Logic

C Core Service Implementation Listings

This appendix provides the specific code implementations referenced in Section **??**: **??**. These listings illustrate the critical logic handling performance, data integrity, and analytical processing within the Digital Shadow platform.

C.1 API Gateway: Request Piping Logic

```
// Dynamic Proxy Implementation using Request Piping
async function proxyRequest(req, res, targetUrl) {
  // 1. Construct the destination URL
  const url = new URL(targetUrl + req.url);

  // 2. Create the forwarding request configuration
  const options = {
    hostname: url.hostname,
    port: url.port,
    path: url.pathname + url.search,
    method: req.method,
    headers: { ...req.headers, host: url.host }
  };

  // 3. Establish the "Pipe"
  // Data flows: Client -> Gateway -> Microservice
  const proxyReq = request(options, (proxyRes) => {
    // Pipe the response back: Microservice -> Gateway -> Client
    res.writeHead(proxyRes.statusCode, proxyRes.headers);
    proxyRes.pipe(res);
  });

  // Start the flow
  req.pipe(proxyReq);
}
```

Listing 6: API Gateway Streaming Proxy Logic (src/api-gateway/api-gateway.js)

C.2 Real-Time Service: Caching Strategy

```
// API endpoint to get all location data
app.get('/api/locations', async (req, res) => {
  try {
    const currentTime = Date.now();

    // Check if we need to refresh the data (Cache Miss or Expired)
    if (!locationDataCache || (currentTime - lastFetchTime) >
    REFRESH_INTERVAL) {
      // Trigger external API request
      locationDataCache = await fetchLocationData();
      lastFetchTime = currentTime;
    }

    // Return valid data from memory (Cache Hit)
    res.json({
      locations: locationDataCache,
      lastUpdate: lastFetchTime,
      count: locationDataCache.length
    });

  } catch (error) {
    console.error('Error fetching location data:', error);
    res.status(500).json({ error: 'Failed to fetch location data' });
  }
});
```

Listing 7: Caching mechanism implementation in the Real-Time Charger Service

C.3 Real-Time Service: Data Normalisation

```
// Simplify complex API status into consistent UI states
let overallStatus = 'Unknown';

if (evses.length > 0) {
  // Get unique statuses from all connectors
  const uniqueStatuses = [...new Set(evses.map(e => e.status.toUpperCase())
    )];

  if (uniqueStatuses.length === 1) {
    overallStatus = uniqueStatuses[0]; // Uniform status
  }
  // Solution for nesting: Explicitly handle mixed availability
  else if (uniqueStatuses.includes('AVAILABLE') && uniqueStatuses.includes(
    'CHARGING')) {
    overallStatus = 'MIXED';
  }
  // Priority fallback for other combinations
  else {
    if (uniqueStatuses.includes('AVAILABLE')) {
        overallStatus = 'AVAILABLE';
    } else if (uniqueStatuses.includes('CHARGING')) {
        overallStatus = 'CHARGING';
    } else {
        overallStatus = evses[0].status;
    }
  }
}
```

Listing 8: Data Normalisation Logic (src/realtime-map-service/realtime-map-service.js)

C.4 Database Sync: Parallel Ingestion

```
// Break large data sync into parallel "batches"
// to maximise throughput without hitting timeout limits
const batchSize = API_CONFIG.BATCH_SIZE;
const batches = Math.ceil(remainingPages / batchSize);

for (let batchIndex = 0; batchIndex < batches; batchIndex++) {
  const batchPromises = [];

  // Launch multiple requests simultaneously
  for (let page = batchStart; page < batchEnd; page++) {
    batchPromises.push(fetchApiData(fromDate, toDate, page * limit));
  }

  // Wait for the entire batch to finish before moving to the next
  const batchResponses = await Promise.all(batchPromises);
}
```

Listing 9: Parallel Data Ingestion Strategy (src/database-sync-service/server.js)

C.5 Data Analytics: Dynamic Query Construction

```
// Build SQL query dynamically based on active filters
// This prevents fetching unnecessary data from the database
const queryParts = [`
  SELECT * FROM charging_sessions
  WHERE start_time >= $1 AND start_time <= $2
`];
const params = [startDate, endDate];

// Only add 'charger_id' filter if user selected a specific charger
if (chargerId) {
  params.push(chargerId);
  queryParts.push(`AND charger_id = $${params.length}`);
}

// Combine parts into final executable SQL
const finalQuery = queryParts.join(' ');
const result = await pool.query(finalQuery, params);
```

Listing 10: Dynamic SQL Query Construction (src/data-dashboard-service/server.js)

C.6 Heatmap Service: Data Transformation

```
// Transform raw database rows into WebGL-ready intensity points
const heatmapData = result.rows.map(row => {
  const totalMinutes = parseInt(row.total_charging_minutes) || 0;

  return {
    coordinates: [
      parseFloat(row.coordinates_lng), // Longitude
      parseFloat(row.coordinates_lat)  // Latitude
    ],
    // The "Weight" for the heatmap intensity
    weight: totalMinutes,
    // Metadata for tooltips
    info: {
      sessions: parseInt(row.total_sessions),
      avgDuration: Math.round(row.avg_session_duration)
    }
  };
});
```

Listing 11: Heatmap Data Transformation (src/heatmap-service/server.js)

C.7 Anomaly Detection: Inference Pipeline

```
def predict_cluster(self, session_data):

    # 1. Feature Extraction (Must match training pipeline exactly)
    # Extracts: duration, hour_of_day, day_of_week
    features, feature_dict = self.prepare_session_features(session_data)

    # 2. Hybrid Inference (Running both models)
    # Scalers ensure input units match training data distribution
    kmeans_scaled = self.kmeans_scaler.transform(features)
    iforest_scaled = self.iforest_scaler.transform(features)

    # 3. Decision Logic
    # Flag as suspicious if EITHER method detects an issue
    is_suspicious = (
        (cluster_id == self.suspicious_cluster_id) or
        (anomaly_prediction == -1)
    )

    return {
        'cluster_id': int(cluster_id),
        'is_suspicious': bool(is_suspicious),
        'risk_level': is_suspicious ? "HIGH RISK" : "NORMAL"
    }
```

Listing 12: Feature Consistency and Inference Logic (src/session_classifier/session_classifier.py)