Technische Hochschule Ingolstadt

Faculty of Electrical Engineering and Information Technology

Master of Engineering - International Automotive Engineering

Master's Thesis

Data-Driven Failure Analysis of Charging Communication

Author:	**Arav Barot**
Matriculation No:	**00122314**
First Examiner:	**Prof. Dr. rer. nat. Armin Arnold**
Second Examiner:	**Dr. Prof. Andreas Hagerer**
Supervisor of Thesis Project:	**Ms. Theresa Gündling & Mr. Ahmed Khliaa**

Acknowledgments

I would like to provide credits of my eductational journey to my Mom and Dad, whose constant encouragement and unwavering support have been my pillars of strength throughout this endeavor. Their belief in me has been a driving force in reaching this milestone.

My sincere appreciation goes to my supervisors at P3 Automotive GmbH, Ms. Theresa Gündling and Mr. Ahmed Khliaa, as well as Prof. Dr. rer. nat. Armin Arnold and Dr. Prof. Andreas Hagerer of Technische Hochschule Ingolstadt. Their continuous support and guidance have been instrumental in allowing me to work independently during my thesis. Their kindness and courtesy in providing the necessary assistance for my progress have been invaluable. Special thanks go to Ms. Maren Lüdke and Mr. Christian Kohl for their unwavering support during my thesis work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

EV	Electric Vehicle
OEM	Original Equipment Manufacturer
DC	Direct Current
AC	Alternating Current
SchuKO	Schutzkontakt (Protactive Contact)
CCS	Combined Charging System
ChaDeMo	Charge De Move
IC	Internal Combustion
EAFO	European Alternative Fuel Observatory
RFID	Radio Frequency Identification
GPS	Global Positioning System
CPO	Charge Point Operator
IC-CPD	In cable control and protection device
PLC	Power line communication
ISO	International Organization for Standardization
DIN	Deutsches Institut für Normung
SOC	State of Charge
V2G	Vehicle to Grid
EVSE	Electric vehicle supply equipment
IT	Information Technology
MSP	Mobility Service Provider
CDR	Charge Detail Record
EMOCH	E-mobility operator Clearing House
QR	Quick Response
EVCC	Electric vehicle charge controller
PWM	Pulse width modulation
IPv6	Internet Protocol version 6
XML	Extensible Markup Language
JSON	Javascript Object Notation
REST API	Representational State Transfer Application Programming Interface
HTTP	Hypertext Tranfer Protocol
BMS	Battery Management System
LED	Light Emitting Diode
FMEA	Failure Mode & Effect Analysis
QTA	Qualitative Trend Analysis
PCA	Principal Component Analysis
DM	Data Mining
ABS	Antilock Braking System
ESP	Electronic Stability Program
VIN	Vehicle Identification Number
ROC	Receiver Operating Characteristics
AUC	Area Under the Curve
B2B	Backend to Backend

Chapter 1

Introduction

The share of Electric Vehicles (EVs) and Plug-in Hybrids has increased over the last few years, and it is growing at a good pace. With significant technological growth in automobile and mobility domain, various types of electric vehicles are introduced in market. Which ranges from Battery Electric vehicle (BEV), Fuel cell Electric vehicle (FCEV) to Hybrid Electric Vehicles and Plug-in Hybrid Electric Vehicles (PHEV). Scope of this thesis is limited to Battery Electric Vehicles and further mentions of EV is referred to the same. On a global level, the expectation of EV adoption to reach 45 percent under currently expected regulatory targets. In the article provided by McKinsey & company it is stated that, "EVs would need to account for 75 percent of passenger car sales globally by 2030, which significantly outpaces the current course and speed of the industry" [1]. The tremendous amount of research in E-mobility, initiatives, and policies from governments, and the growing public charging networks had increased the popularity of EVs. These vehicles do not contribute to noise pollution, and they are locally emission-free when the grid is connected to renewable sources of energy making Mobility more sustainable.

Currently, Electric Vehicles are positioned at a higher cost when compared to vehicles with Combustion engines in a similar category.Where margins permit, OEMs are lowering prices, which increases affordability and acts as an additional stimulant for increased sales.[2]. There are several factors that influence the cost reduction. Socio-Political acceptance, Market acceptance, Matured technology, mass production, demand, and developed infrastructure.

The EV charging infrastructure is growing day by day. Convenience in charging of EV is one of the major factors that influences the Market acceptance, infrastructure, and demand for EVs. As the charging system of an EV works very differently than refuelling the IC engine vehicle, the vehicle operator needs to adapt to the new process psychologically. The reliable EV charging infrastructure plays an essential part to attract more people towards using EVs and keep them using them. As EVs take a considerable amount of time to charge and error & interruption-free charging process is expected by the user.

Chapter 2

Fundamentals of E-mobility & Charging Ecosystem

In this chapter, various methods of charging an EV are explained. Furthermore, types of connector for wired charging method are explained as well. To provide a thesis context types of public charging facility, Charging modes are also explained. This chapter further provides an overview of the entire charging ecosystem of the present day and its evolution, including various entities in it. This chapter also provides information about communication protocols and various charging initiation methods.

2.1 Methods of charging EV

The charging ecosystem in Europe is evolving rapidly[3]. Subsequent to developments in mobility several methods of recharging EV are proposed or available in the current mobility market, which can be wired and wireless methods of recharging the battery or swapping out the battery.

In urban E-scooters-like vehicles, swapping out the batteries or removing them for charging is a common method. The primary advantage of battery swapping technology lies in its ability to accelerate the process of replacing a drained battery with a fully charged one, resulting in significantly reduced time requirements. In recent years, a few automotive companies have introduced concepts of vehicle battery swapping facilities. The battery pack, which is located beneath the structure of the vehicle, is quickly accessible and can be simply replaced. In this concept, the process of recharging with a charged battery pack is accomplished within a short span of time. The battery pack is a modular idea that may be adjusted according to the user's preferences in terms of performance, efficiency, or range parameters[4]. However, this strategy needs to cope with inventory concerns to maintain the necessary numbers of batteries in reserve and requires huge investments to develop the parallel infrastructure of a few specific types of vehicles.

There is another method for charging EV, a wireless charging method. The wireless charging in electric vehicles operates in a manner similar to the wireless charging devices used to charge common household devices, such as mobile phones and smartwatches. In this charging method, both the car and the charging pad are fitted with conducting coils. The technology used involves the utilization of resonant electromagnetic induction. The induction of current flow in a closed circuit of a receiver coil is a result of the changes in electromagnetic flux caused by the transmitter coil of a charging pad[5]. Wireless charging is also further categorized into two modes. Static wireless charging where wireless charging can be done when the vehicle is static or parked. Dynamic wireless charging enables charging of the vehicle while it is moving, allowing EV users to charge the vehicle without taking breaks for charging. However, inductive charging is efficient when the distance between the transmitting coil and the receiving coil is placed in proximity. This technology is still in the standardization phase and not commercially available[6].

In the current market, wired charging stands as the dominant method for charging EVs. Where an EV can be charged using a pantograph or a plug-in cable. Electric locomotives have been using pantographs for a considerable period of time. The connection between the overhead cables and the pantograph can be created using either a retractable or permanently connected mechanism[7]. The utilization of pantograph charging can be considered convenient in cases where the electric vehicle's route is pre-established. This particular charging method has been implemented by a few of European towns for the purpose of facilitating local public bus transportation. In recent times, there has been a concept of electric long-hauling trucks equipped with overhead retractable pantographs that establish connections to high-voltage direct current (DC) lines.

The utilization of a Plug-in cable for charging an EV is a typically used method. This form of EV charging involves the provision of a dedicated charging port within the vehicle, thus allowing for the connection of a separate charging cable. This cable serves to establish a connection between a charging port on one end and charging equipment on the other end, enabling the transmission of electric power.

2.2 Types of Connectors used in EV for charging

The diversity of charging plug types in the EV Market is contingent upon the specific charging method employed by each individual EV. The selection of plug type installed in an EV is determined by various criteria, including the permissible charging speed, permissible charging current type, government regulations and market considerations. The many types of connection plugs are specified and categorized in accordance with the DIN EN 62196 standard. The Current type, Power and voltage of the widely used plug connec-

tors are illustrated in the table 2.1 . These connectors are offered in two configurations.

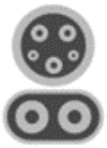
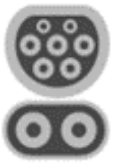
Image of the Connectors					
Name of the Connector	Type 1	Type 2	CCS 1	CCS 2	ChaDeMo
Charging type and Maximum Power output	AC, 120V Single Phase – 1.6kW / 240V Single Phase – 19.2 kW	AC, 230V Single Phase - 7.5kW / 400V Three-Phase – 22kW	DC, 1000V 360kW	DC, 1000V 360kW	DC, 500V 400kW

Table 2.1: Types of Plug Connectors

One with a permanent attachment to the charging station, and another with a separate cable that can be detached and stored in a vehicle. The detachable cable option features a SchuKo (Schutzkontakt – Protective Contact) type connector at the opposite end. Certain charging stations are additionally fitted with SchuKo-type connectors commonly found in households. Users of the EV can connect their own detachable cable to initiate the charging process, utilizing single-phase alternating current (AC) with a maximum power capacity of 2.3 kilowatts (kW)[8].

2.3 Classification and Accessibility of EV Charging Facilities

In contrast to internal combustion engine (IC) vehicles, which require going to gas stations for refueling, EV users have the advantage of being able to recharge their vehicles at multiple facilities. The present-day EV charging infrastructure provides a diverse range of charging options to the user. This charging infrastructure includes home charging via standard plugs or wall boxes, strategically integrated charging facilities within enclosed parking areas of commercial establishments such as workplaces and malls, as well as dedicated charging stations. These facilities can be categorized into three broad classifications. These are 1. Private charging places, 2. Semi-Private charging places, and 3. Public charging places.

Private charging places

In order to utilize private charging facilities for EVs, it is necessary for EV users to obtain authorization from the owner of the charging point. Typically, these charging facilities

include charging points situated at residential homes, privately owned parking lots, and designated parking areas for apartment complexes.

Semi-private charging places

These charging sites provide access to charging exclusively for a specific group of customers. Furthermore, the proprietor of the charging location possesses a certain level of authority over the area. Semi-private charging facilities encompass various sites, including office parking facilities, charging stations situated within university premises, and dealership establishments. These locations impose specific conditions, such as restricting access to only office employees, university students, or individuals who own a particular brand of EV. Other users are explicitly prohibited from utilizing these charging facilities for their vehicles in Semi-private charging places.

Public charging places

The public charging facilities are referred to charging places in a public area, where regardless of ownership of the particular area, the charging point is available for all EV users. These charging facilities include charging points available at roadside parking, city parking plots, shopping centers and malls and public charging stations. The growth in publically available charging stations is 175% in Europe between 2020 and 2023 as per the report from the European Alternative Fuel Observatory (EAFO)[9]. This thesis primarily focuses on the EV charged at public charging stations.

2.4 Connectors, Charging Points, and Charging Stations

Further EU Sustainable Transport Forum provides a description of publicly available charging infrastructure[9]. The architecture of the charging station and entities which are included in the charging station are illustrated in 2.1

Connector

A connector refers to a physical interface that facilitates the transfer of electric power from the charging station to an electric vehicle.[9]. The Connector comprises a cable features a female-type connector at one end, designed to connect to a car, and a male-type plug at the other end, intended to be plugged into a port at a charging point. A pantograph, an induction plate, or a permanently attached plug at the charging point is typically utilized for fast charging.

Charging Point

The transmission of electric power occurs via a charging point. A charging station could include one or many connectors to accommodate various connector types, although only one can be utilized concurrently. The following further explains it: There are the same

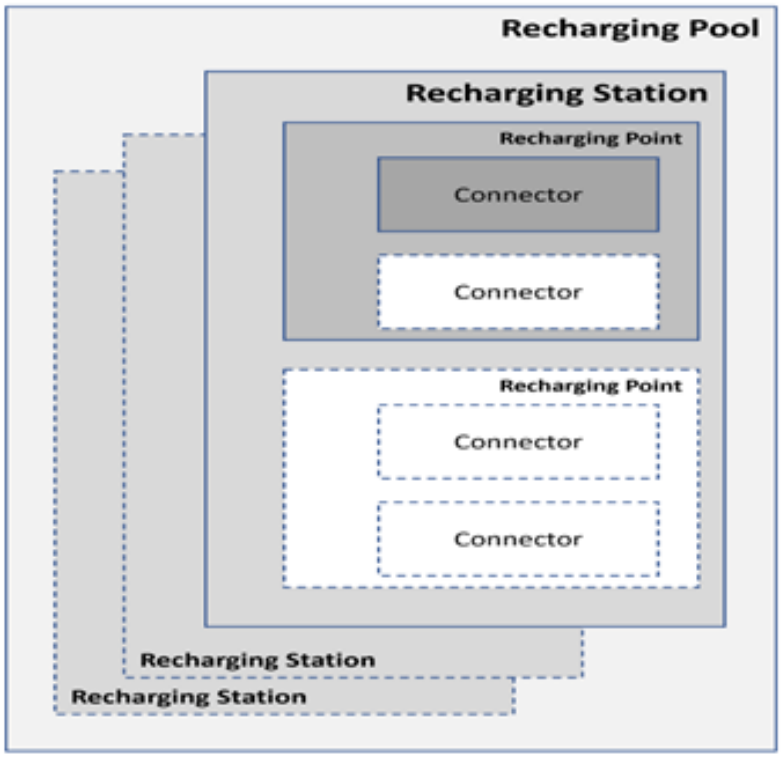


Figure 2.1: Nested architecture of Charging pool, station, charging point, and connector as per EAFO [9]

number of spaces for parking and charging points per charging station[9].

Charging Station

A charging station is a physical entity that consists of one or several charging points, which are equipped with a common user identification interface. The physical interfaces between humans and machines, such as RFID readers and displays, are typically situated at the charging station. Some charging stations that offer only plug-and-charge functionality do not require an input interface[9].

Charging Pool

A charging pool consists of one or many charging stations and associated parking spaces. The charging pool is managed by a single Charge Point Operator (CPO) situated at a specific location with corresponding GPS coordinates. The charging pool is a cartographic element that symbolizes the charging infrastructure represented on a map.[9]

2.5 EV Charging Modes

When it comes to the charging speed of the charging station and EV, it is categorized as normal and fast charging, which has a close relation with the type of input current provided to EV by a charging station. The levels of charging of an EV are described in detail inside the DIN EN 61815-1. These are categorized as charging Mode 1, Mode 2, Mode

3, and Mode 4 respectively.

Mode 1

This mode facilitates the charging process using a standard household plug on the charging station side. A single or three-phase AC current at 230V with a maximum charging power of 3.7kW can be used as the input current. There is no provision for communication between the infrastructure and a vehicle. Input current and voltage using this charging method are controlled by an onboard charger controller fitted in a vehicle. This charging method is primarily utilized in light electric vehicles such as E-Scooters.

Mode 2

In mode 2, it is possible to communicate with the charging infrastructure In Cable Control and Protection device(IC CPD) and the electric vehicle. Other aspects that differentiate mode 1 from mode 2 include the fact that in charging mode 2, a car can be charged at a maximum charging power of up to 22 kW at 230V AC. Due to more charging power vehicle is charged faster compared to mode 1. However, this method is also considered as a normal speed charging.

Mode 3

The electric vehicle (EV) is charged in mode 3 using the power plug that is supplied with the car. However, it is necessary to have a permanently installed charging device, such as a wall box charger or charging station, in order to charge the car. In this particular charging mode, the vehicle has the capability to be charged using either a 230V or 400V power supply, with the option of utilizing either AC single phase or three phase current. The charging capacity of this mode extends up to 43.5kW. Furthermore, in this operational state, the vehicle establishes communication with the infrastructure through Power line communication (PLC) in accordance with the ISO 15118 communication requirements.

Mode 4

DC fast charging is the common term for mode 4. Permanently constructed high power charging infrastructure is needed to charge EVs in this mode. The cables utilized for the transmission of power from the charging point to the EV are equipped with a liquid cooling system. High level communication between the car and infrastructure is necessary when charging a vehicle using DC power. Additionally, it adheres to the ISO 15118 requirements.

Charging an EV with mode 1 to 3 using AC current is practically slower. The power is delivered during the process slowly. This results in comparatively flat charging characteristics. The flatter AC charging profile is illustrated in 2.2 The reason behind the flat and slow charging is the small size of the AC to DC converter in the vehicle. While

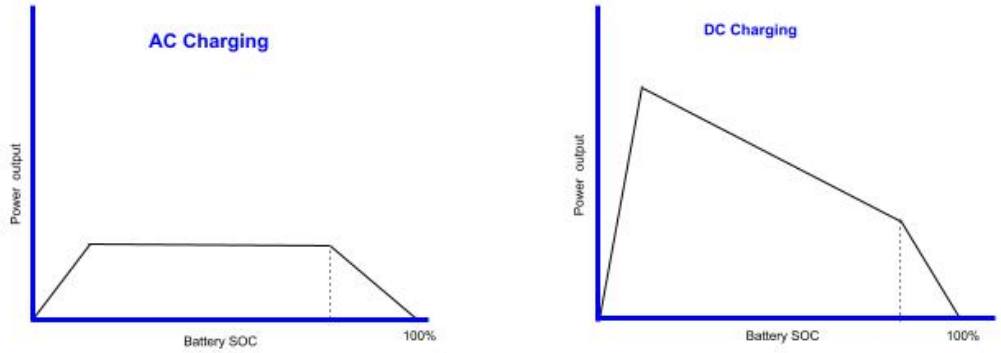


Figure 2.2: Charging profile comparison between AC and DC vehicle charging

charging with DC in mode 4 is significantly faster and charging behaviour of the EV is quite different. In the ideal condition the input power quickly peaks and then gradually slows down as vehicle gets charged as illustrated in DC charging profile in 2.2. As SOC of vehicle increases ability to safely handle incoming power reduces. When vehicle is preconditioned for DC power input, it can be charged even with higher power resulting faster charging. Longer duration of charging with high power heats up charger and battery components. To keep battery in optimum performing range active cooling of battery and power components are required. However, the exact charging strategy varies from model to model of EV which is pre-determined by a manufacturer in order to optimize the charging speed, maintain battery health and to make charging process safer.

2.6 The Evolution of Electric Vehicle Charging Ecosystem

During initial period, EVs and EV users were bound to several constraints such as limited mode of payment, availability of charging services with specific charge point operators, and lack of connected and real-time monitoring features from mobile apps. The involvement of 4G/5G network technology in vehicles has allowed manufacturers to provide connected car features to the end users where they can monitor the vehicle charge status and other parameters remotely and further concept of V2G has been introduced. This gradual evolution in the infrastructure of E-mobility can be categorized into 3 distinct phases.

2.6.1 Evolution Phases of EV charging ecosystem

Phase 1

In the first phase, EV owners could only charge their vehicles with local public EVSE of a Charge point operator with which had a running contract. During this phase charging

options at public charging stations were severely limited. Which had restricted users from planning long journeys using an EV.

Phase 2
In the second phase with the expansion of EVs, the charging network happened, where mobility service providers and aggregators entered a market that increased the interoperability of the EVSE for cross-country hassle-free charging with the concept of roaming.

Phase 3
The third or current phase, is recognized by massive growth in EV charging infrastructure and the entry of traditional energy businesses, payment service providers, connected app services, and so on. As the charging ecosystem is maturing various players are entering such as grid load management, home energy management, and vehicle-to-grid as bi-directional charging. These changes are the reason for the evolution of the smart charging ecosystem.

With each phase, the complexity of the ecosystem and its underlying IT landscape grows substantially. The major role player s in the backend IT landscape of the current phase infrastructure are in 2.3. In addition to the new market entrants, charging solutions from AC wall boxes to high-power chargers are being developed rapidly leading to a massive increase in the diversity of market solutions for EV charging.[10]

2.6.2 Actors and Entities in current Phase EV charging Ecosystem

Modern-day charging ecosystem as explained in 2.6.2 includes complex IT infrastructure to fulfill various demands. These actors and entities are connected to each other through connections such as contractual relations, expectations, exchange of information, exchange of power or both. 2.3 illustrates essential actors and entities that play important roles in a modern-day Charging ecosystem. The roles of these entities are explained in 2.2.

These actors and entities communicate in the backend to provide a user facility for charging and other side services such as charging process monitoring and billing. It is essential to have established communication gateways 2.4 between actors.

User
The user is generally the owner of the vehicle or a fleet owner who has made a contract for EV charging and side facilities with the Mobility service provider (MSP). In this particular scenario, the expectation of user is to get the EV charged without any hassle.[11] Once charging process is finished irrespective of intentional or unintentional stop of the current flow, user receives bill generated by MSP, on the bases of Charge detail record(CDR).

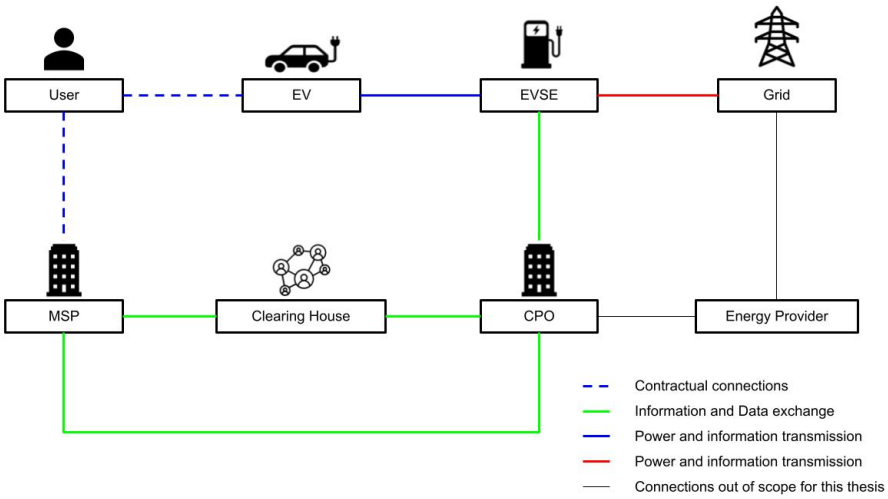


Figure 2.3: A Schematic overview of Electric Mobility Interconnectivity

CPO

Charge point operator(CPO) provides the charging infrastructure of the charging pool to charge the EV. Managing, installing, and maintaining charging stations of the responsibility of CPO. CPO is also responsible for generating invoices.

MSP

Mobility service provider(MSP) felicitates user with various mobility services. User signs a two party contract with the MSP in advance using charging services. MSP provides the end user with the RFID card to charge the vehicle at charging station and provides several app based services as well. This services includes, Remote starting of the charging services, locating charging stations and so on. The user receives the bill for the usage of charging services from MSP.

EMOCH

E-mobility operator clearing house (EMOCH) provides an IT-infrastructure, that communicates between CPO and MSP. Clearing houses are also referred as aggregator. To provide the end user the hassle free charging experience, eRoaming comes into play. EMOCH receives dynamic and static data from the CPO and sends further data which are Point of Interest for MSP. Clearing houses follows certain standard protocol to aggregate data such as requests, CDR and KPI. Aggregator also provides additional services to CPO such as charging plot reservation and charge scheduling. 2.2 aggregates the essential roles of actors in the Charging ecosystem.

Entity	Role
EVSE	Electric vehicle supply equipment, often known as charging points transmits energy to an EV. Each public charging point has a unique EVSEID. Sometimes one charging station can have multiple EVSE ID to distinguish multiple charging points available at the station.
CPO	Charge points operators manages and provides charging networks and deliver, install, and maintain charging points. CPO determines the the prices for charging an EV as well. CPO has an active interface with the energy provider.
EMOCH	E-Mobility clearing houses act as data providers from CPO to MSP or vice-versa when they do not have a standard or peer-to-peer data transfer facility.
MSP	Mobility Service Providers provide mobility services and products such as billing subscriptions and payment methods such as RFID or through an APP to end users.

Table 2.2: Roles of Entities and Actors in the E-mobility charging communication

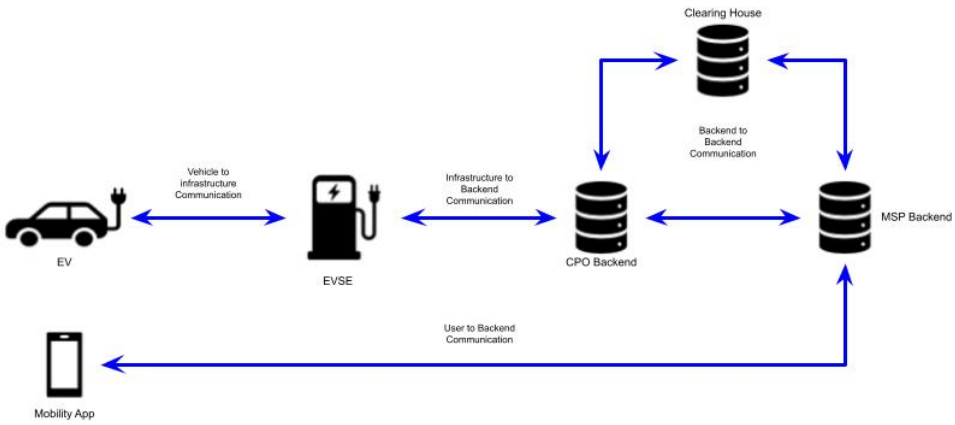


Figure 2.4: Overview of backend communication between various actors in charging infrastructure

2.7 Methods and modes of initializing charging service:

In the current phase of EV charging infrastructure 2.6.1, there are several methods to initiate charging services after plugging in the charging cable. The options of initiating the charging process provide flexibility to the user and enhance user experience with the charging process. Here these initialization tools are identified as Clients. Once an EV user plugs in the charging cable with the vehicle, these methods enable user to start the charging process. These methods are as follows.

1. RFID Card
2. Remote / App clients
3. Plug & Charge

RFID Card

The RFID cards allow users to start with charging service by tapping the card scanner at the EVSE. Authentication and request to charge with an RFID card is the most commonly used and reliable method to initiate charging of an EV. Generally, the EV user initiates charging with tapping the RFID card on the scanner mounted on the EVSE 2.5. The EVSE initiates the authentication request in this method, which is then transmitted to the CPO. The received request is thereafter transmitted to MSP either through a clearing house or via peer-to-peer communication. Once the authorization from MSP has been approved. MSP grants permission to initiate the charging process.

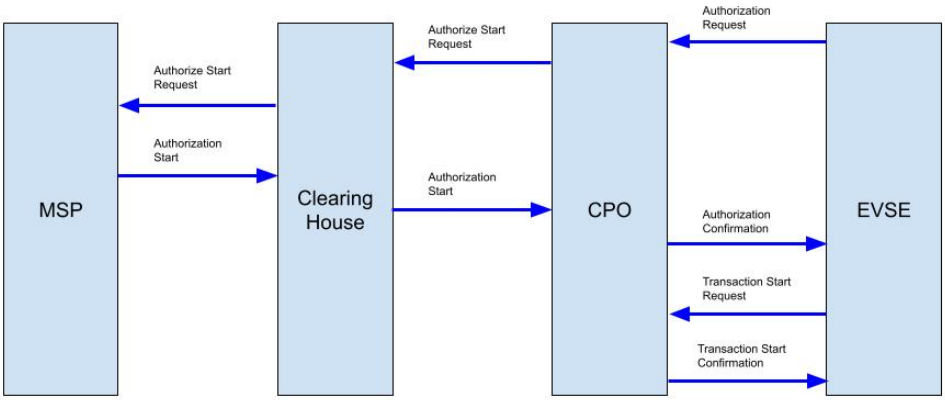


Figure 2.5: Message communication chain when RFID is used to initialize charging process

Remote / App Clients

When this method is used, the initiation of the charging service is done by either a Remote service or an MSP mobile phone application 2.6. The start request is transmitted from the user's mobile device to the MSP. The remote start request is transmitted from the MSP of user, to CPO via the clearing house. In this case MSP only sends remote start charging request to further actor after a user gets authenticated. After receiving a remote start request, the EVSE initiates the charging process and sends an acknowledgment message to the App clients over the same chain of actors.

The mobile app facilitates users, the reservation of available charging ports within its interface. The user has the choice to either scan the QR code provided on the EVSE or manually choose the EVSE or charging outlet through the app. Once the request has been authorized, connect the plug-in device for charging.

Plug & Charge

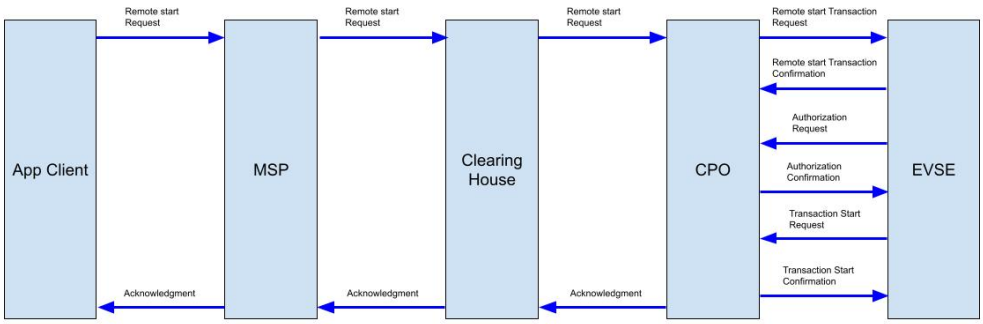


Figure 2.6: Message communication chain when charging process is initialized through remote or app clients

Plug & Charge enables automated communication and invoicing procedures between electric vehicles and charging stations, adhering to ISO 15118 standards and ensuring secure data communication. The plug and charge method is a new method of charging that has been recently introduced to the market. The Electric Vehicle Charge Controller (EVCC) technique involves the initiation of a charge request and the authentication of identity by the vehicle 2.7. The Plug & Charge function is designed to minimize user effort and offer a seamless charging experience. With this feature, users can simply connect their charger without the need of any additional authentication methods.

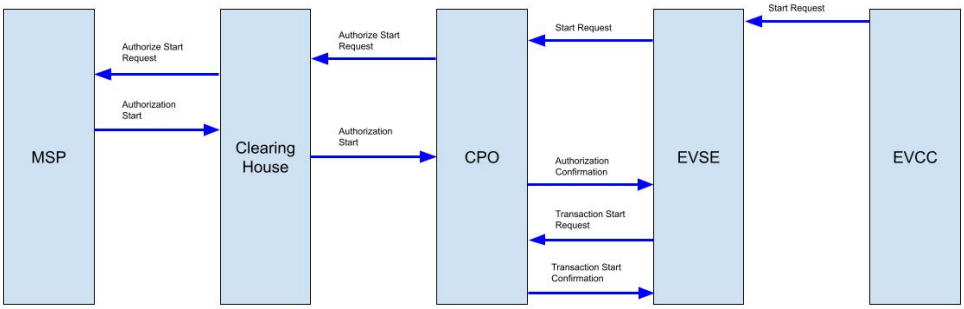


Figure 2.7: Message communication chain when Plug and charge method is used to initialize charging process

Once the charging cable is connected and locked up, and communication between the electric vehicle (EV) and electric vehicle supply equipment (EVSE) is established, the charging station initiates the collection of all required data for the charging process. The transmission of data continues even during the process of charging the vehicle. The charging station produces a Charge Record Details (CDR) that contains all essential data required for the invoicing process. The charging station further transmits the CDR to the operator. The charge invoice is generated using the CDR and subsequently transmitted to the Mobility Service Provider (MSP). The billing amount and requisite data are sent to the end user by the MSP.

2.8 eRoaming

With the present phase of EV Charging infrastructure 2.6.1 interoperability and eRoaming are important and necessary features available. With eRoaming, different entities can communicate with each other to provide a seamless charging experience to the EV user. eRoaming services are more or less the same as using roaming services in mobile phones. In the charging ecosystem, it provides a facility for the user to charge the EV with the other operator with which user does not have signed a contract. In this case, the clearing house receives data from the operator where the EV is getting charged and it is further forwarded for the billing process to the MSP which has a contractual relationship with the user.

Definition of eRoaming
When it comes to e-mobility, roaming refers to the ability of an electric vehicle (EV) driver under contract with a Mobility Service Provider (MSP) to charge at a location run by a Charge Point Operator (CPO), with whom the MSP has a contract but not directly with the EV driver. This can be done directly or through a roaming hub.[12].

2.9 Communication protocols between various actors in Charging ecosystem

To enable eRoaming 2.8, it is crucial to ensure interoperability between actors within the charging ecosystem. Irrespective of the manufacturer and service provider, these entities should have the capability of communicating crucial signals and messages with one another. In order to have effective communication, the sender must transmit a message that can be effectively comprehended by the receiver. To address this problem, several protocols and communication standards have been developed to establish a uniform framework for communication.

Communication between EV and EVSE
When an EV is connected to a charging cable, there is an establishment of the communication channel. Through the communication channel vehicle charging communication controller and supply equipment communication controller communicate. In this case, the communication channel between vehicle to infrastructure is used to follow EIC-61851 standards. This standard enabled to establishment of lower-level information exchange between the vehicle and EVSE. In this communication, data exchange is possible by Pulse width Modulation (PWM).

Currently, the lower-level data exchange is not sufficient to provide services such as fast

charging, load management, Bi-directional charging, and smart connected services. In the current market, there are a number of EV manufacturers, CPO operators, and EVSE manufacturers with numerous models in their portfolios. Therefore, there is a strong requirement for standardized communication methods and messages that can be sent and interpreted by systems without error. This communication is standardized by ISO 15118 (Road Vehicles: Vehicle to grid communication interface) which uses Powerline Communication (PLC) to communicate. EV communicates requests such as requests to initiate charging, stop charging, throttle the input power, temperature of the battery, State of charge and health, Input power, and so on with the EVSE during DC charging. ISO 15118 features with secured IPv6-based client-server approach. Digital communication adhering to this standard is done through compressed XML format between EV and EVSE. ISO 15118 also supports plug-and-charge authorization by using several cryptographic methods to authenticate the identity of the EV user and, user-specific requirements such as reliability, availability, error handling & reporting.[13] This ISO standard also emphasizes the importance of developing a system that prioritizes ease of use for end users. This standard mentions utility-specific requirements like communication for safety, protection from overcurrent and voltage, and power transfer limitations for load and grid energy balancing.[14]

Communication between CPO, MSP, and EMOCH

Several widely used communication protocols in the backend of various charging system infrastructures are mentioned in 2.3

No.	Communication Protocol
1	Open Charge Point Protocol (OCPP)
2	EIC 63110 – Protocol for management of electric vehicles charging and discharging infrastructures
3	Open Inter Charge Protocol (OICP)
4	Open Clearing House Protocol (OCHP)
5	eMobility Inter-Operation Protocol (eMIP)

Table 2.3: Widely use backend communication protocols in the charging infrastructure.

In these protocols OCPP and ISO 63110 are protocols for communication between the charging station and the charging station management system. While OICP, OCHP, and eMIP are protocols for communication between CPO, MSP, and Clearing houses like B2B backend communication systems.

Open Charge Point Protocol & EIC 63110

This literature refers OCPP 2.0.1 version released on 03.2020. Any kind of charging technique can be used with the industry-supported Open Charge Point Protocol (OCPP), which is the de facto standard for communication between a charging station and a charging station management system (CSMS). OCPP is an open standard with no cost or li-

censing barriers for adoption[13]. The basis of OCPP is a JSON/RESTful API, with WebSocket serving as the primary communication protocol. The Charge Point plays the role of a WebSocket client, whereas the Central System is the WebSocket server[14].

This communication protocol is widely accepted by Charge Point Operators to communicate with the Charging stations or EVSE. This protocol manages the backend data of the charging station and EVSE in a standardized way. Moreover, it monitors and communicates CDR and billing data as well. The referred version of the OCPP also includes most of the functionalities of the previous version of OCPP 1.6. There are changes in the name of messages in OCPP 2.0.1, however, the functionality of the messages remains the same. The OCCP 2.0.1 protocol provides features such as plug and charge, support of display showing details such as consumed energy and prices, and remote operation of the charging process such as remote start and stop of the process[13].

EIC 63110 is recently (07.2022) published international standard for communication between EVSE and CPO. This standard also addresses features such as management of energy transfer, grid usage, information exchanges related to the energy requirement, contractual data, authentication and authorization, eRoaming, pricing and so on. This standard also covers the requirement of cyber security and encryption of data for the exchange[15].

Open Intercharge Protocol

This literature refers OICP 2.2 version released on 03.2018. OICP is an EV roaming protocol that standardizes communication between MSPs and CPOs [14]. This protocol is developed by Hubject Brokering systems. Hubject GmbH is a joint venture between the BMW Group, Bosch, Daimler, EnBW, RWE, and Siemens.[16]. OICP is also based on JSON/RESTful API using HTTP as a basic communication protocol, which is real-time and supports both synchronous and asynchronous operations[14].

Open Charge Point Interface

This literature refers OCPI 2.2.1 version released on 10.2021. The Open Charge Point Interface (OCPI) supports scalable and automated EV roaming between Charge Point Operators and eMobility Service Providers.It enables authorization, charge point information exchange (including live status updates and transactional events), charge detail record exchange, remote charge point commands, and smart charging-related data communication across parties.[17]. OCPI is built on JSON/RESTful API and uses HTTP as its primary communication mechanism. It is a real-time protocol that enables both synchronous and asynchronous activities.[14].

eMobility Inter-operation Protocol

This literature refers eMIP 1.0.7 version released on 07.2019. eMIP protocol is managed by GIREVE founded by EDF, Renault, CNR, and Caisse des Dépôts. It is based on SOAP with HTTP as the basic communication protocol. Despite being built as a real-time protocol, eMIP supports asynchronous operations.[14].

Chapter 3

Problem Statement

As stated in 1, the electric vehicle requires a significant amount of time to be recharged. This scenario is also applicable when the electric vehicle is charged using public charging stations. When electric vehicles are plugged in for the charging process at a public or semi-public charging station, there are a number of actors who play crucial roles in facilitating charging and other facilities associated with charging. It is necessary for these actors to communicate with one another and share information during the entirety of the charging process in order to initiate the process of charging the vehicle, to continue charging the vehicle, or to terminate the charging process 2.6.2. During the charging process, when these actors communicates with eachother, the data is logged in the databases of CPO and MSP. This data flows through different channels and uses different protocols for communication 2.9. Furthermore, there are a number of internal parameters of an electric vehicle that are monitored and communicated in terms of signals during the charging process in order to ensure that the charging process is carried out in a secure manner. These parameters are also logged by MSP and relevant information are logged by CPO as well in order to create CDR. The temperature of the battery and the battery management system (BMS), the current and voltage levels at the input and the BMS, the charging power, the state of charge, and the state of health of the high-voltage battery are some of the main parameters that are monitored in order to regulate the current flow. The logged data in the overview represents the condition and features of the vehicle, details of the charging equipment, geolocations, time stamps of various events, channels of communications, and the record of occurred errors.

Automotive manufacturer and service providers can gain valuable insights into the charging behavior of vehicles under various ambient and intrinsic conditions of the vehicles as well as with different types of charging systems and communication protocols. This information is advantageous for them to troubleshoot problems in the present situation and to take a retrospective look while developing new systems.

The failure that happens during the charging process can be attributed to a number of

different combinations of characteristics, each of which has the potential to create issues with the charging process.

Error detected but charging continued
This type of error occurs when the charging of the vehicle is continued, however, there has been an error occurred in any of the systems.

Error detected and charging stopped
This type of error occurs when the charging of the vehicle is interrupted due to the error occurring in any of the systems. The charging interruption indicates that charging power has been unexpectedly stopped for unintentional reasons during the process

Error shown on the LED
This type of error is logged in the database when the LED light on the EV has been blinked or lit up indicating that there is an error in the charging process. During this error charging of the vehicle maybe continued or interrupted.

Scope of the Thesis

- Electric vehicles (EVs) represent a significant advancement in sustainable transportation, but their widespread adoption hinges on effective charging infrastructure and reliable operation during the charging process. However, understanding the diverse charging patterns, behaviors, and potential errors encountered by EV at different charging stations remains a challenge. In this thesis the Dataset has been obtained from the database of a vehicle manufacturer through P3 Automotive GmbH which includes a data of a specific model of the EV.

Aim of the Thesis

- This thesis seeks to methodically analyze and characterize EV charging behaviors, focusing on identifying key performance indicators (KPIs) that may indicate the presence of errors in the charging process. By discerning these patterns, it is aimed to establish a foundation for enhancing the reliability of the charging ecosystem.
- What KPIs have the most significant influence on errors during the EV charging process, or play a pivotal role in error identification, thus facilitating improvements in the reliability?
- Exploratory analysis of the dataset to identify patterns in overall data with regards to the charging behaviour.

Chapter 4

Data Driven Analysis

In this chapter, an overview of the fault analysis method is provided initially. Further, a detailed exploration of the thesis context covers methods of data-driven analysis using machine learning algorithms, including supervised and unsupervised algorithms, along with statistical methods. These methods encompass Clustering, Random forest, and PCA. Additionally, types of data and data quality are explained to comprehend the use cases of machine learning algorithms.

4.1 Fault analysis Approaches

Fault detection and analysis is a crucial component of process monitoring, and it plays an important part in the study of reliability engineering as a tool for determining when and where failures and faults have occurred. As per ISO/EIC 2383-14:1997 fault is defined as, An abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function[18]. While failure is defined as, The termination of the ability of a functional unit to perform a required function[18]. Early detection of the fault in the system helps to prevent additional harm and decreases the chance of a loss of development, production, economy, and life. Fault detection measures are mainly categorized in two different approaches. Which are Model-based and Data-driven subsequently. These Fault diagnosis approaches consist of several structured frameworks of addressing faults in systems.

Model Based analysis Method

In this form of analysis method, the system is represented by schematics or mathematical functions 4.1. The model is developed to be a representation of the real system, and it is expected that it shall include all of the possible information regarding the system that could be formulated for an accurate diagnosis. After the model has been created, various combinations of input parameters, which can be vectors or scalar signals are fed into a model so that the defect can be detected or simulated[19]. Thus model-based fault detection systems rely on predefined models of system behaviour to identify deviations or

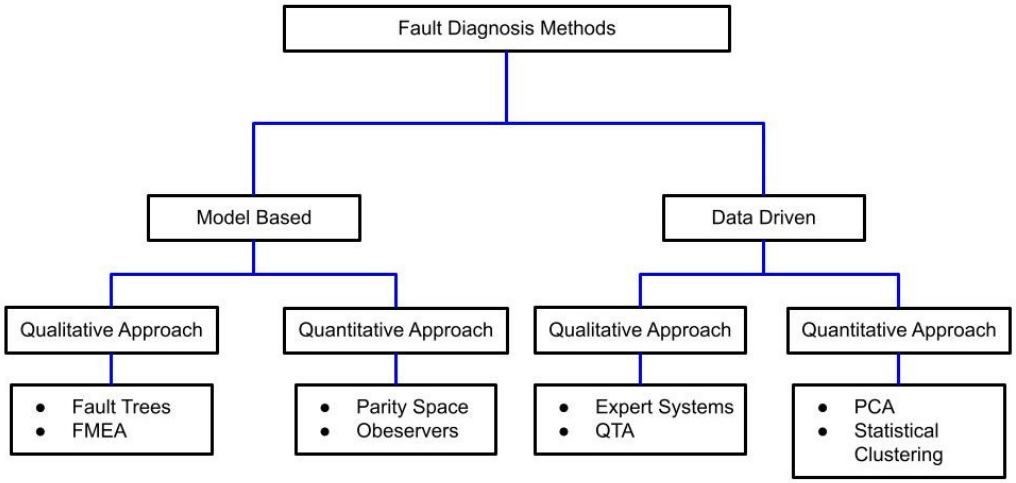


Figure 4.1: A Tree diagram providing an overview of different types of Fault Diagnosis methods

abnormalities.

Data-Driven analysis Method

A large amount of historical data has been taken in order to perform data-driven analysis method 4.1. By using various methods data is transformed into useful knowledge[20].

As technological advancements continue to make systems more intelligent, the processes that should be carried out to achieve the required outcomes in industry are getting increasingly complicated. There are many possible combinations of parameters or functions that can lead to certain failures or results that fall short of expectations. It is difficult to spot these patterns of failure and fault using model-based techniques since it is extremely difficult to incorporate all parameters and constraints into a model and therefore several assumptions are taken. This makes it difficult to recognize these patterns of failure and fault. In this circumstance, the use of quantitative and data-driven approaches is required to locate or estimate potential failure spots. Data-driven fault detection methods offer distinct advantages in handling the intricate and dynamic nature of modern industrial systems. A continuous feed of the historical data makes Data-driven method more adaptable to complex and evolving processes.

4.1.1 Types of Data

In the current business environment, companies gather massive quantities of data in their database management systems, including logs, data from sensors, and numerous activities[21]. These data are further used for many purposes such as proof of records, improvement and development programs and troubleshoot or diagnosis of existing problems.

A comprehensive knowledge of the type of data being used for analysis is crucial. The choice of techniques for data analysis and machine learning algorithms that will be used are significantly affected by the type of data. The following list of datatypes includes several types of data.

1. Categorical Data
2. Numerical Data
3. Spatial Data
4. Multivariate data

Categorical data

Categorical data is defined as information "relating to names." The name, symbol, or sign is related to these categories of data properties. The categorical data consists of both non-numeric and numeric data. The terms "Charging type," "Country," and "Error code" are a few instances of this datatype[22].

Numerical data

Numerical data are quantifiable and can be represented as integers or real numbers. Consumed Energy, Charging Duration, and Battery Temperature are some examples of this datatype[22].

Spatial Data

Spatial data includes information on the objects or vectors that make up a space. In general, it is used to represent geographical locations and location-related information like "Latitude and Longitude"[23].

Multivariate data:

Data that incorporates more than two different types is known as multivariate data. This data type includes a mix of categorical, numerical, and/or geographical data types. This type of data is also referred to as mixed data in data mining algorithms.

4.2 Data Quality

Throughout a data lifecycle, data passes through many stages 2.4. Following are the general flow steps. Data generation, data acquisition, storage, and analysis constitute all steps in the data lifecycle. During data generation quality issues such as noise, inconsistency, unreadability, and false measurements are occurred. When data is received and stored in the database there are chances of loss of information, alteration of the values and formats, duplication of data points, and change in the precision points of data. This issue causes problems in data analysis by producing unreliable and inaccurate results. A data quality

check must therefore be performed before the data is utilized for further analysis.
In a quality assessment of data, 6 criteria are considered which are summarized in the illustration 4.2.

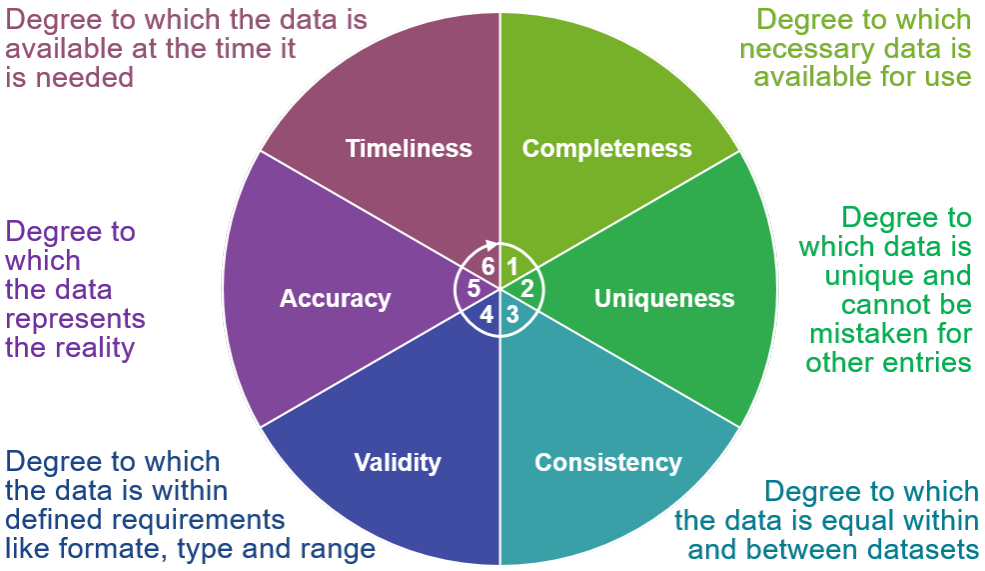


Figure 4.2: Six essential characteristic to evaluate quality of Data

Handling of Missing data

When data is stored in the database, there are chances that some crucial information may be missing.To further analyze the data or to feed it into algorithms for analysis, data imputation has been performed. Data imputation involves filling in missing values in the dataset to maintain consistency in the analysis. It is essential to understand the underlying causes of missing values before employing imputation methods to populate the vacant data points for data analysis. The lack of values does not necessarily imply their insignificance. The presence of missing values in a data-driven analysis may suggest the occurrence of an issue or scenario that compromised the data collection process. Instead of imputing anticipated values, it is recommended to identify missing data for the purpose of outlier or pattern recognition. In the present situation, the missing values are addressed by the utilization of the arbitrary value imputation approach.
In the field of feature engineering, it is observed that arbitrary values fall beyond the range of the feature. For instance, in cases where the feature range is positive, a suitable imputation value could be -1.

4.3 Data Mining using Machine Learning

The process of extracting useful knowledge, patterns, trends, or valuable insights from large amounts of data is known as data mining (DM) or knowledge discovery in data (KDD). It is a method that involves the use of various methodologies or algorithm to derive valuable insights from raw data. Data Mining is a subset of data analysis and is closely associated with machine learning and statistics.

There are several methods used widely for knowledge discovery in the field of data science including Graph theoretic methods like 1. Breadth-first search and depth-first search, 2. Dijkstra's Algorithm for the shortest path 3. Minimum Spanning tree and statistical methods such as 1. Principal Component Analysis, 2. Classification, Clustering and Regression models[24].

Plenty of the useful insights that data provides are difficult to find using conventional statistical techniques and take a lot of time to study. Machine learning algorithms come in quite handy in this situation. Machine learning algorithms are commonly used for data analysis, including data mining and information collection. They are excellent at recognizing parameter combinations that could result in system failures or detecting behavioral patterns within a specific system.

This thesis focuses on data-driven analysis using statistical methodologies and machine learning. In the field of artificial intelligence, machine learning is used to find patterns and relationships in datasets that are meaningful rather than depending on manually defined rules[25]. It usually needs little or no human involvement.

The machine learning approach differs from the classical rule-based algorithm 4.3. Classical programs use a set of clearly defined rules to achieve a specific objective. While machine learning makes rules and approximations based on the mapping between the input and the expected class or cluster.

Figure 4.3: A comparitive illustration of Classical programming algorithms and Machine learning algorithms

In general, the learning process in machine learning models are iterative process and can be classified into the following categories. 1. Supervised machine learning, 2. Semi-

supervised machine learning, and 3. Unsupervised machine learning.

Supervised machine learning

Liu and Wu state the definition of supervised machine learning as "Supervised Learning is a machine learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. As the output is regarded as the label of the input data or the supervision, an input-output training sample is also called labeled training data, or supervised data"[26]. The goal of supervised machine learning is to minimize the misclassification or the error between the target and computed output. The paradigm of supervised machine learning algorithm is illustrated in 4.4.

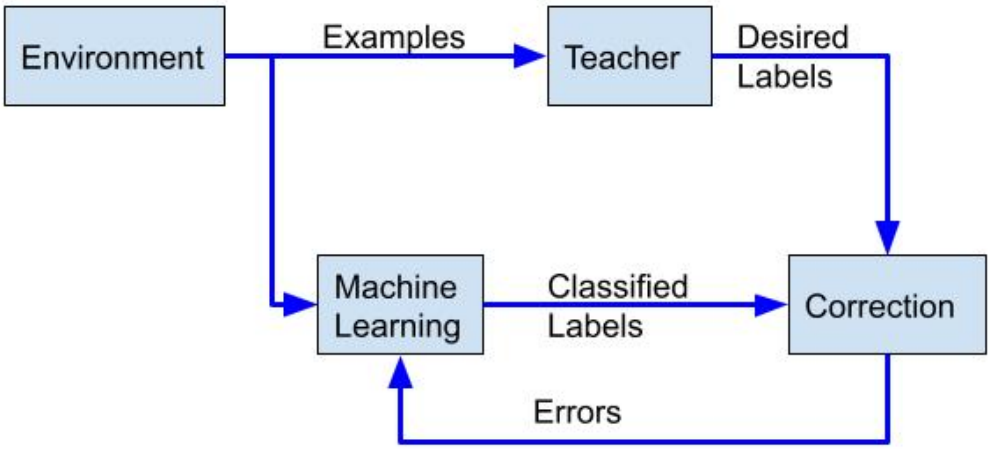


Figure 4.4: Representation of a learning and working methodology of supervised machine learning algorithms

Unsupervised machine learning

The paradigm of unsupervised machine learning is illustrated in 4.5. These types of machine learning algorithms are capable of recognizing patterns without the use of training labels. This method uses every vector in the set as an input for the analysis. Unsupervised learning aims to map out the data's distribution within the input space and cluster them. This type of machine learning algorithm are mainly used for clustering the association of data. The interest in unsupervised learning is to get structural information from the data frame.

4.4 Clustering Techniques

Cluster analysis is a range of multivariate techniques that are mainly used to group items according to their inherent features. Clustering, in essence, involves the act of assembling

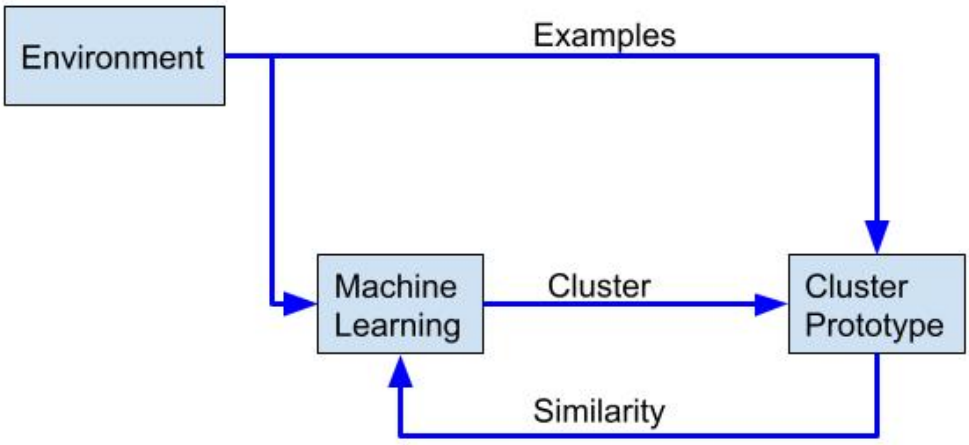


Figure 4.5: Representation of working behaviour of unsupervised machine learning algorithms

comparable data points together based on specific qualities or attributes. The main objective of clustering is to recognize the inherent structure inside data, revealing relationships and associations that may not be immediately clearly apparent. By identifying these, clustering provides a way to understanding underlying phenomena in dataset. Clustering fundamentally relies on the notion that data points belonging to the same closer demonstrates higher similarity among them compared to data points in different clusters. There are various clustering methodologies available. However, the choice of clustering algorithm is determined by the nature of the data, the desired outputs, and the challenges present in the dataset.

Figure 4.6: Association and Grouping of data using Clustering algorithm

4.4.1 K-Means Clustering Algorithm

MacQueen and Anderberg proposed the K-means clustering algorithm, which is one of the widely recognized and used non-hierarchical unsupervised machine learning methods. This method is an approach to grouping the data from the dataset into non overlapping clusters. When applying this method, each data point in the dataset is assigned to a particular cluster. K-means is a type of crisp clustering, in which each data point is assigned to only one cluster. In the name of K-means, K is the number of defined clusters[27].

The K-means method is formulated in as following mathematical problem P:
Minimize,

$$P(W,Q) = \sum_{i=1}^{k}\sum_{i=1}^{n} W_{i,l}d(X_i,Q_i)\text{[28]} \tag{4.1}$$

Subject To,

$$\begin{aligned} \sum_{l}^{k} w_{i,l} = 1, \quad 1 \le i \le n \\ w_{i,l} \in 0,1, \quad 1 \le i \le i \le n, l \le 1 \le k \end{aligned} \tag{4.2}$$

[28]

where, W is an $n \times k$ partition matrix, $Q = \{Q1, Q2, ..., Qk\}$ in 4.1 is a set of objects in the same object domain, and $d(.,.)$ is the distance measure between two objects. Example of used distance measures are given in 4.5 and 4.6. in 4.2, $w_{i,l} \in W$.

Problem P in 4.1 can be solved by iteratively solving the following two problems:
1. Problem $P1$: Fix $Q = \widehat{Q}$ and solve the reduced problem $P(W, \widehat{Q})$
2. Problem $P2$: Fix $W = \widehat{W}$ and solve the reduced problem $P(\widehat{W}, Q)$
Problem $P1$ is solved by:

$$\begin{aligned} w_{i,l} = 1 \quad if \quad d(X_i,Q_t) \quad for \quad 1 \le t \le k \\ w_{i,t} = 0 \quad for \quad t \ne l \end{aligned} \tag{4.3}$$

[28]

Problem $P2$ is solved by:

$$Q_{l,j} = \frac{\sum_{i=1}^{n} w_{i,l}x_{i,j}}{\sum_{i=1}^{n} w_{i,l}} \quad for \quad 1 \le l \le k, \quad and \quad 1 \le j \le m\text{[28]} \tag{4.4}$$

The algorithm to minimize $P(W,Q)$ is an iterative process. values of $w_{i,l}$ and $q_{l,j}$ are gained iteratively after an initiation of clustering algorithm using 4.3 and 4.4

The distance measure d in 4.1 is commonly employed as a metric for assessing simi-

larity between datapoints. Greater value of distance matric indicates the lack of similarity between those data points. Similarity represents the degree of correspondence among all the data points that are used in an analysis.

Euclidean Distance: Euclidean distance is the distance between points in a straight line. The distance matrix uses Pythagorean theorem[29].

$$d(x_i, q_l) = \sqrt{\sum (x_i - q_l)^2} \tag{4.5}$$

Euclidean distance is the most used distance measure in clustering. However, for several applications Manhattan distance is used as well.

Manhattan Distance: Manhattan distance is the sum of absolute differences along each dimension[29].

$$d(x_i, q_l) = \sum |x_i, q_l| \tag{4.6}$$

In the flowchart 4.7 iterative approach of K-means clustering algorithm is illustrated. Where K objects from the dataset are chosen from a dataset as cluster centeroids. Then each data point is assigned to the cluster based on the mean value. After that means of clusters are updated and this iterative process is done until the group converges and does not move. the value of P in 4.1 gets lower with each iteration and at the convergence point for the selected numbers of clusters K the value of P is the lowest.

4.4.2 K-Modes Clustering Algorithm

Categorical data is clustered using the K-Modes algorithm, which is an extension of K-means 4.4.1. The K-modes uses dissimilarity measure, in contrast to K-means, can solve the P2 problem with categorical data. The K-modes approach for categorical objects uses a simple matching dissimilarity measure. When solving the problem P2, the algorithm replaces the cluster means with modes and uses a frequency-based technique to find the modes. Huang formulates K-modes algorithm as a Problem P.
To Minimize,

$$P(W, Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,l} \delta(x_{i,j}, q_{i,j}) \text{[28]} \tag{4.7}$$

Ultimately summations in 4.7 can be broken into distinct equations in 4.8 and 4.9. while the delta function is represented as 4.8.

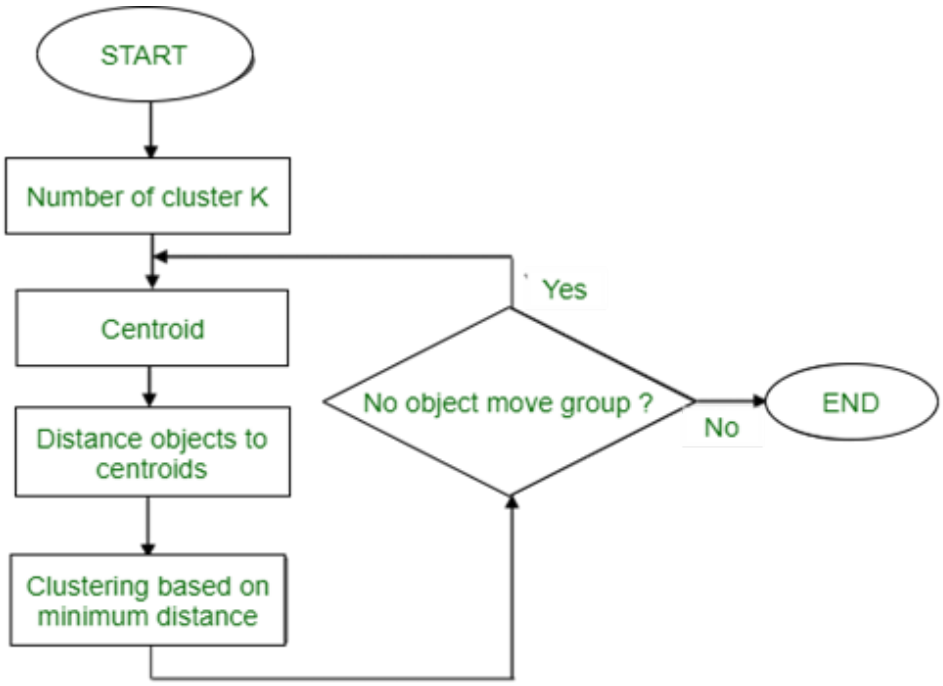


Figure 4.7: Flow chart explaining an iterative approach of K-means clustering

Dissimilarity Measure

Let X, Y represent two category objects with m categorical properties. The dissimilarity measures between X and Y are determined by total mismatches in the respective attribute category of two objects. The lesser the number of mismatches, the more comparable the two things. Kaufman and Rousseeuw presented the basic matching measure[30], which is as follows,

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \tag{4.8}$$

where,

$$\delta(x_j, y_j) = \begin{cases} 0 & \text{when } (x_j = y_j) \\ 1 & \text{when } (x_j \neq y_j) \end{cases} \tag{4.9}$$

Mode of set

X categorical object is described by categorical attributes, $A_1, A_2, \ldots, A_m$.
A Mode of $X = X_1, X_2, \ldots, X_n$ is a Vector $Q = [q_1, q_2, \ldots, q_m]$ that minimises,

$$D(X,Q) = \sum_{i=1}^{n} d_1(X_i, Q) \tag{4.10}$$

In 4.10, Q is not necessarily an element of X. Finding a Mode of set: Let $n_{j,k}$ be the number of objects having the k^{th} category $c_{j,k}$ in attribute A_j and $f_r(A_j = c_{k,j}|X = \frac{n_{c_{k,j}}}{n})$ the relative frequency of category $c_{k,j}$ in X.

Huang mentions a theorem to define a way to find Q for given X. The theorem states that,

The function $D(X,Q)$ *is minimised iff* $f_r(A_j = q_j|X) \geq f_r(A_j = c_{k,j}|X)$ *for* $q_j \neq c_{k,j}$ *for all* $j = 1,\ldots,m$

4.4.3 K-Prototypes Clustering Algorithm

K prototypes algorithm is developed to cluster mixed datatypes. This algorithm integrates K-means 4.4.1 and K-Modes 4.4.2 into K-prototypes. When this algorithm is used dissimilarity measure used to calculate distance is the weighted sum of distance measure used in K-means and K-modes algorithms.

The dissimilarity between two mixed-type objects X and Y, which are described by attributes $A_1^r, A_2^r, \ldots, A_p^r, A_(p+1)^c, \ldots, A_m^c$, can be measured by,

$$d(X,Y) = \sum_{j=1}^{p}(x_j - y_j)^2 + \gamma \sum_{j=p+1} \delta(x_j + y_j) \tag{4.11}$$

In the equation 4.11 the first part is the Euclidean distance between data points, while the second term is the simple dissimilarity function. γ is a weight to eliminate the bias in attributes.
Using the mixed type objects, P can be modified as:

$$P(W,Q) = \sum_{l=1}^{k}(P_l^r + P_l^c) \tag{4.12}$$

Where,

$$P_l^r = \sum_{i=1}^{n} w_{i,l} \sum_{j=1}^{p}(x_{i,j)} - q_{l,j})^2 \tag{4.13}$$

and

$$P_l^r = \gamma \sum_{i=1}^{n} w_{i,l} \sum_{j=1}^{p}(x_{i,j} - q_{l,j})^2 \tag{4.14}$$

The Summation of equations 4.13 and 4.14 results in,

$$P(W,Q) = \sum_{l=1}^{k}(\sum_{i=1}^{n} w_{i,l} \sum_{j=1}^{p}(x_{i,j} - q_{l,j})^2 + \gamma \sum_{i=1}^{n} w_{i,l} \sum_{j=p+1}^{m} \delta(x_{i,j}, q_{l,j}))[28] \tag{4.15}$$

Since P_l^r and P_l^c in 4.15 both are non-Negative, minimizing $P(W,Q)$ is equivalent to minimizing P_l^r and P_l^c for $1 \leq l \leq k$.

4.4.4 Hierarchical Clustering

The hierarchical clustering method in contrast to flat clustering methods, constructs a hierarchical structure that represents the similarity between elements in the dataset[31]. Hierarchical clustering is often represented using a binary tree structure, in which data elements are stored in the leaves. The two most similar data points are then selected and kept in closest to each other, subsequently merging the two nodes. This binary tree structure graphically visualized in terms of Dendrogram 4.8.

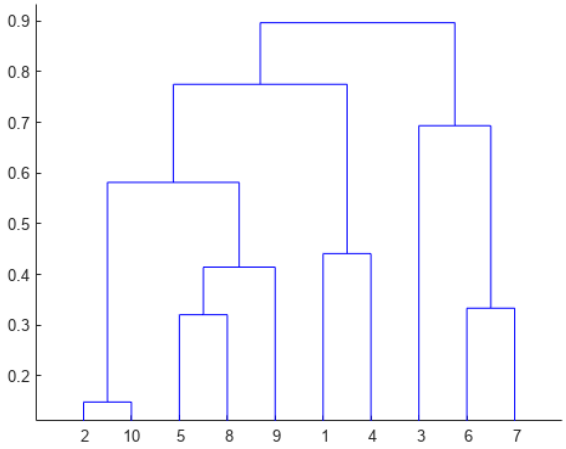


Figure 4.8: A visual representation of Hierarchical clustering using Dendrogram where features on X axis and Distance on Y axis

The described illustration 4.8 of the Dendrogram refers to the data points on which the clustering method is applied. The data points are arranged in a manner where the two most similar data points are positioned adjacent to each other. These data points are linked through a node. The node's height corresponds to the degree of similarity among the datapoints. The degree of similarity in Hierarchical clustering is determined by measuring the distance where these nodes are merging. Hierarchical clustering involves the merging of nodes in pairs until a root node is reached, which contains all elements and forms a hierarchy.

When Euclidean distance is chosen as a base to measure the distance between two elements of X, and the minimum distance as the linkage for defining the sub-set distance $\Delta(X_i, X_j) = min_{x \in X, y \in Y} D(x,y)$.

Linkage Distance

Linkage distance in hierarchical clustering is the measure of dissimilarity or distance between clusters. The linkage method defines how the distance between clusters shall be measured. The choice of linkage method influences the merging process when hierarchies are forming. There are various types of linkage methods available to use in hierarchical clustering. Popularly used linkage distance methods are illustrated in 4.9. In which, two different clusters are shown with datapoints inside represented as dots.

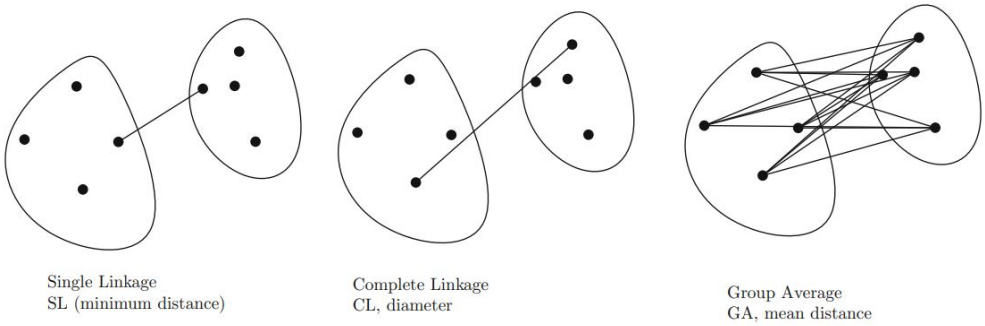


Figure 4.9: Linkage methodologies used in Hierarchical clustering: Single Linkage, Complete Linkage & Group Average Linkage

Single Linkage: Single linkage distance is the shortest distance between any two points in the two clusters.

$$\Delta(X_i, X_j) = min_{x_i \in X_i, x_j \in X_j} D(x_i, x_j) \tag{4.16}$$

Complete Linkage: Complete Linkage is the maximum distance between any two points in the two clusters.

$$\Delta(X_i, X_j) = max_{x_i \in X_i, x_j \in X_j} D(x_i, x_j) \tag{4.17}$$

Group Average Linkage: Group Average Linkage is defined as the average distance between all pairs of points, were one point is from the first cluster and the other is from the second cluster.

$$\Delta(X_i, X_j) = \frac{1}{|X_i||X_j|} \sum_{x_i \in X_i} \sum_{x_j \in X_j} D(x_i, x_j) \tag{4.18}$$

Ward's Method: This method uses the technique of minimizing the increase in variance when two clusters are merged.

$$\Delta(X_i, X_j) = \frac{n_A n_B}{n_A + n_B} \cdot D^2(X_i, X_j) \tag{4.19}$$

where , D in 4.19 is the increase in the variance when clusters X_i and X_j are merged into cluster X_k. n_i and n_j are the sizes of clusters X_i and X_j, respectively. $\Delta(X_i, X_j)$ represents the dissimilarity between two clusters. While from 4.16 to 4.18 D is the distance metric eg. Euclidean Distance 4.5 or Manhattan Distance 4.6.

Principal component Analysis

Principal component analysis is a statistical method to reduce the dimensionality of the observation space. This method is used when there is a need of bringing down large numbers of features into few meaningful features and it also reflects the correlation and interdependencies of features in the dataframe. The reduction is obtained by creating a linear combination of the observation space. These linear combinations are principal components. These principal components are orthogonal to each other and capture most of the variance in the data. In 4.20 and 4.21 PCA is applied to find the major axis of data in 2D.

Given a set of 2D data points, $(x_1, y_1), \ldots, (x_n y_n)$, X is taken as a random variable of the first component and Y as a second component of the 2D vector. The covariance matrix of components is defined as,

$$M(X,Y) = \begin{vmatrix} Cov(X,X) & Cov(X,Y) \\ Cov(X,Y) & Cov(Y,Y) \end{vmatrix} \tag{4.20}$$

Where,

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y})) \tag{4.21}$$

In 4.21 $\bar{x}$ and $\bar{y}$ are mean of datapoints of components.
Computation of Eigenvalues and Eigen vector to identify Principal Components:
Let M be a square $n \times n$ matrix, P be a non-zero eigenvector and λ be an eigen value of which,

$$MP = \lambda P \tag{4.22}$$

If the formula is rearranged,

$$(M - \lambda I)P = 0 \tag{4.23}$$

The principal axis of the data is the eigen vectors of the data's covariance matrix, and

the projection of the data's instances onto these principal axes are known as the principal components. The next step is to reduce the number of dimensions by keeping only the axes that provide the most variance and eliminating the rest.

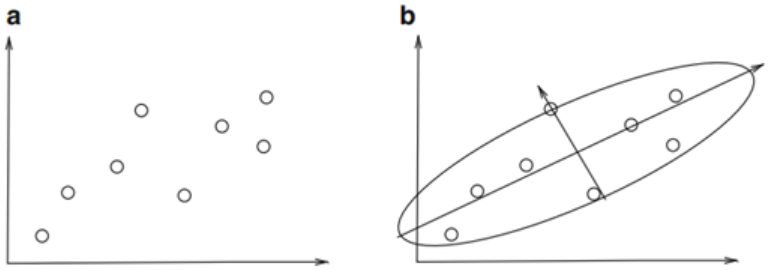


Figure 4.10: Illustration of Derivation of an Eigen Vector for Principal Component Analysis

4.5 Decision Tree and Random Forest

4.5.1 Decision Tree

A Decision tree is a highly practical method for classifying data. It is a hierarchical model consisting of discriminant functions or decision rules that are applied repeatedly to classify the data into smaller subsets[32]. The decision tree utilizes a rule-based technique to generate an output. These subsets are partitioned until the stopping criterion is met[33]. Since decision trees are constructed using predefined classification labels, they fall under the category of supervised machine learning models.

A decision tree is composed of multiple branch and leaf nodes. The example of decision tree is illustrated in 4.11. Every node symbolizes a characteristic of a dataset. The starting point of a decision tree is referred to as the root node. The node to which the following nodes are connected is known as the parent node, while the connected nodes themselves are termed offspring nodes. A branch node is formed when a node is connected to succeeding nodes that include properties of the dataset. The node from which the output is anticipated is referred to as a leaf node. There are two primary methods for constructing a Decision Tree. These two used primarily methods for constructing decision trees are ID3 and CART.

ID3 - Iterative Dichotomiser

This algorithm is developed by Ros Quinlan[34]. The ID3 algorithm employs a top-down technique to generate a decision tree. This approach utilizes a greedy algorithm that picks features and estimates the information gain associated with each feature. The attribute with the greatest information gain is selected as the root node. Alternatively, arcs represent more potential values. Then, every instance of outcomes are checked to see whether

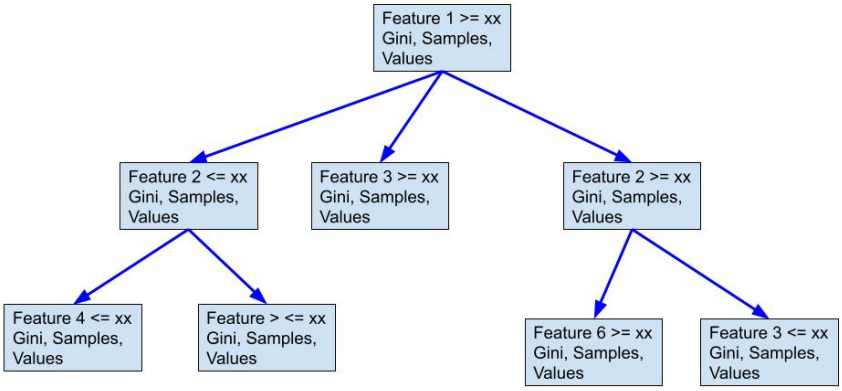


Figure 4.11: Binary decision Tree with Leaf, Branch and Root node and made with 6 exampled features

they belong to the same class or not. ID3 algorithm only supports categorical values for constructing this particular decision tree[35].

CART - Classification and Regression Tree

Classification and Regression Trees are introduced by Breiman[36]. A classification tree is built by the process of binary splitting, where a feature is divided into two distinct groups. The Gini index is used as a criterion for splitting attributes. The CART algorithm has the capability to manage missing values in a dataset by automatically incorporating surrogate splits[37]. CART is compatible with features that have non-linear correlations with one another. The feature space is split up in a recursive manner based on the input features, which results in the creation of non-linear decision boundaries.

Node Splitting

At each node T of the decision tree, features j are chosen. The procedure for selecting the feature will be elaborated upon in the following paragraphs. Following the selection of a feature, it next chooses the threshold value s of the particular feature and divides the data into two subsets. Here R_{left} and R_{right} denotes subsets of data points that go to left and right nodes respectively, a split at node T.

$$R_{left}(T) = \{X|X_j \leq s\} R_{right}(T) = \{X|X_j > s\} \tag{4.24}$$

Decision Boundary

Recursive splitting is a procedure that results in the development of a tree structure that contains decision bundaries. When a feature is divided, each split expresses a condition on the feature, which results in nonlinear decision boundaries in the feature space.

Gini Impurity

Gini Impurity index is the measures the likelyhood that a randomly selected datapoint would be incorrectly classified by a specific node. The abbreviation of Gini stands for generalized inequality index. Gini impurity can be used in building a decision tree. The value of the Gini impurity is between 1 and 0. Elements with perfectly homogeneous and all similar values in the sample have a Gini value of 0. While Gini value 1 means maximum inequality in the elements of the sample. Gini index is a sum of the square of the probabilities of each class[38]. In the process of building decision trees, at the first Gini impurity of all features in dataset is calculated.The feature with the lowest Gini impurity index is designated as the primary node in a decision tree. The process involves selecting a specific class of a parent node as subnodes, followed by calculating the Gini Impurity index of the remaining features. The feature with the lowest Gini impurity is then selected again. The operation is iterated until only a node that is completely pure remains. A pure node is a node that contains just instances of only one class of the feature.

Let, the dataset has n different classes. While p is proportion of a class in the dataset. Proportion of dataset with n classes are $p_1, p_2, ..., p_n$
Thus, the Gini Impurity index is Mathematically represented as,

$$GiniIndex = 1 - \sum_{i=1}^{n} p_i^2 \tag{4.25}$$

Entropy

Other way to build the decision tree is using Entropy. Entropy is a quantitative measure of the randomness of the information being processed. If all data in the feature is homogeneous and all data is similar then the Entropy is 0 while elements are equally divided then the entropy value rises toward the maximum value 1[39]. The decision tree building procedure using Entropy functions in a manner comparable to Gini Impurity.
Entropy here is mathematically represented as,

$$Entropy = -\sum_{i=1}^{n} p_i \times \log_2(p_i) \tag{4.26}$$

Information Gain

Information gain quantifies the amount of information that a feature expresses about the class. Decreased entropy and Gini index result in higher information gain. High entropy and Gini index result in less information gain.Information gain measures the difference between the entropy of the dataset prior to splitting and the average entropy of the dataset after splitting, taking into account the specific feature F with values $v_1, v_2, ..., v_n$ being considered.

$$InformationGain(T,F) = Entropy(T) - \sum_{v \in F} \frac{|T_v|}{T}.Entropy(T) \tag{4.27}$$

Information using Gini impurity can be derived as,

$$InformationGain(T,F) = Gini(T) - \sum_{v \in F} \frac{|T_v|}{T}.Gini(T) \tag{4.28}$$

4.5.2 Random Forest

Random forest, as the name suggests is made of more than one individual decision trees. In this methods Decision trees are made using random sampling with replacements. Using a random forest algorithm both classification and regression related task can be done. Similar to the Decision Tree approach, Random Forest is a type of Supervised Machine Learning algorithm. Random forest provides multiple decision trees which are built differently but trained on the same dataset. As Random Forest is composed of separate decision trees, each tree produces its own distinctive classification output. The output

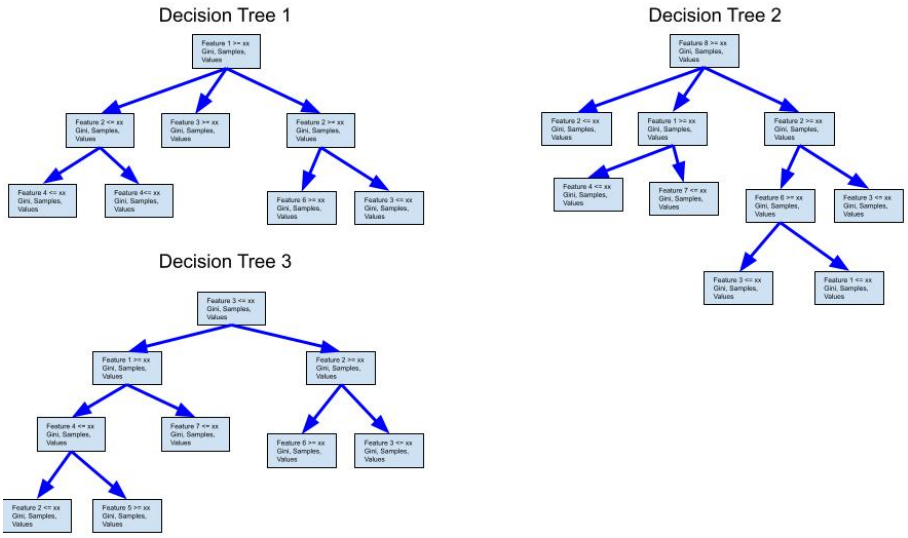


Figure 4.12: Random Forest made with 3 decision trees which are nonidentical to each other by structure and configuration

from each trees is subsequently averaged or assigned a majority ranking based on the particular instance of use.The decision tree and random forest methods are applicable to both categorical and numerical data. However, these algorithms are unable to process datasets that contain a mix of data types. Consequently, it is necessary to convert a dataset into numerical data by the process of one hot encoding. In a below given example of the Random forest there are 3 differently built decisions tree consisting of different features and hierarchy of features from each other. Hence there are different splitting criterion

and decision boundary created. Results from 3 different decision trees are aggregated to generate the final output of the Random Forest.

Chapter 5

General Steps

This chapter offers comprehensive details of the data utilized for the thesis and the methods employed for data analysis. It includes an explanation of the evaluation strategy and hyperparameters of the random forest and K-prototype clustering algorithms. In this chapter brief explaination of tools and language used for the thesis are given as well.

5.1 Description of Data

In this thesis, the Key Performance Indicators (KPIs) are referred to using alternative terms to maintain the confidentiality of the data, as the original names of the KPIs are not disclosed.

The dataset which is used for this thesis is made by joining two different datasets. The 1st dataset consists of various parameters measured by sensors of EV during a charging process and several information about the vehicle configuration. This dataset consists of around 64 different features. These features include the type of battery, the status of charging, and 4 different error codes which are designated by the vehicle in case of various failures. Moreover, minimum, maximum, and average values of charging current, power, battery temperature, BMS temperature, State of Health and charge of high voltage and low voltage battery packs, targeted and actual preconditioning of the battery, outside temperature, speed signal from ABS and ESP, Timestamps of charging events, Difference is state of charge before and after charging, Total consumed power, power used for auxiliary requirements.

On the other hand, the 2nd dataset consists of the data from the EVSE side. This data is recorded by EVSE during the charging process and measurements are taken by sensors from EVSE and the information of EVSE. It contains data such as Consumed energy, Event sequences, time stamps of event sequences, EVSE ID , ChargePoint class, Start request type, error code, error sequence, the platform used for backend communication etc.

These 2 datasets are joined using a common identifier key. In this case the identifier key is the Vehicle Identification Number (VIN). After doing a data modelling and data preparation for analysis, a summary of the dataset is presented below.

The dataset is highly imbalanced as data points with errors are covering around 23% of the whole dataset. The imbalance in the dataset can cause bias in machine learning

Total numbers of data points from EV side	2650443
Total numbers of data points from Charging sessions side	870324
Total Matched datapoints	73752
Points with - Error Detected but charging continued	4237
Points with - Error Detected and charging stopped	2172
Points with - LED Notification Error	5237

Table 5.1: Description of the Dataset

algorithms. In unsupervised machine learning algorithms such as K-means clustering imbalanced datasets have effects such as Centroid Bias, Misclassification of minority classes, Reduced sensitivity to outliers, and bad initialization of centroids. With supervised machine learning algorithms imbalance in the dataset can cause bias during training where the algorithm used recognizes certain classes more accurately and misclassifies the minor classes.

The data quality check, improvement & standardization, and the algorithm development has been done in Python 3.10. With this language the most used open source libraries during the thesis work is mentioned in 5.2

5.2 Tools used

The data quality check, improvement & standardization, and the algorithm development has been done in Python 3.10. With this language the most used open source libraries during the thesis work are mentioned below.

Pandas

This library provides a wide range of tools to handle and manipulate data frames. These include tools to read, write & join datasets, reshape datasets, slice, index, alignment, handle missing data, and many other operations[40]. In the thesis work Pandas are used for dataframe manipulation and creation as well as other operations regarding standardizing data and reorganizing features.

Numpy

Numpy is also an open-source Python library. Numpy is widely used for multidimensional array operations. Moreover, Numpy for this thesis work is used for shape manipulation, logical as well as statistical operations on the large amount of data[41].

Sci-kit Learn

Sci-kit Learn is a powerful library that contains tools for Statistical and Machine Learning operations. Few of many tools that this library offers are, Scaling tools, tool for Principal Component Analysis, and KMeans clustering[42].

KModes from PyPI

Python Package Index (PyPI) is a third-party open source Python repository. This repository allows developers to publish their Python packages and other users to utilize them in individual projects. These uploaded packages on PyPI provide ease of sharing, reusing, and maintaining a code quality.

PyPI contains one of the majorly used packages for this thesis, which is KModes. This repository contains the python python packages which, allow the user to use the extension of KMeans clustering to cluster the categorical and mixed datatype without encoding them into numerical values[43].

SciPy

SciPy is an open-source Python library majorly used for technical and scientific computing. SciPy contains various modules for scientific operations such as integration, interpolation, and ODE Solvers[44]. In the thesis work SciPY library is used for distance calculations for Hierarchical clustering and to derive ranking orders of features.

Matplotlib

Matplotlib is a comprehensive library for creating static and interactive visualizations in Python. This library allows user to create various types of plots such as Bar plots, scatter plots, error bars, pie charts, and so on. These plots can be visualized in either 2D or 3D as per the requirement of the user. The results from the statistical methods and machine learning algorithm during this thesis work are visualized using the Matplotlib library[45].

Data Normalization using MinMax Scaler

Features measured at different scales may not contribute equally to model fitting in machine learning algorithms and PCA, which may lead to bias in its results. Data normalization is thus crucial to solving this issue. When clustering algorithms using Euclidean distance are employed normalization of data is more beneficial[46].

MinMax scaler is the is a very useful data normalization method which can be formulated as follows,

$$x_{scaled} = \frac{x - min(x)}{max(x) - min(x)} \tag{5.1}$$

When this function is applied on data, the range of data points are transformed in decimal points of 0 and 1 where minimum and maximum values get transformed in 0 and 1 respectively.

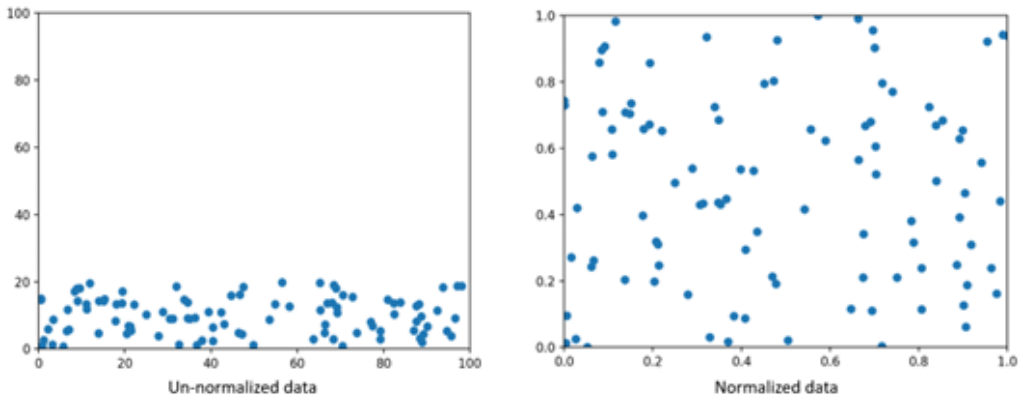


Figure 5.1: Importance of Normalization of data before feeding into machine learning algorithm

5.3 Application of Clustering

The selected method aims to uncover underlying patterns within the combined dataframes, which consist of data from both the vehicle and charging session aspects. The main purpose of using the clustering algorithm is to discover patterns in the overall charging behavior and to observe how the charging process is carried out with varying values of features, as well as to determine which parameters are comparable to or dissimilar to one another in a macroscopic manner. In addition to this, once clusters have been generated, the intention is to conduct a detailed analysis of the clusters and locate anomalies that are associated with the clusters.

In order to utilize the K-Means algorithm, the user must manually select the cluster size K. In the K-Means method, the cluster center serves as the representation of its own cluster. The objective of the algorithm is to identify K clusters while minimizing the overall error. This sometimes introduces a challenge in selecting the optimal number of cluster numbers K, for the subsequent operation.

5.3.1 Selecting correct numbers of clusters

To decide the number of optimum clusters K several metrics are used, which are discussed in this section.

Elbow Plot

The main idea behind the Elbow method is to try the wide range of clusters number k and monitor the reduced overall error in the previously discussed in 4.4.1 $P(W,Q)$. As the number of cluster K increases, the sample assignment will get more precise. The degree of each cluster will progressively increase meanwhile the cost function P will reduce.

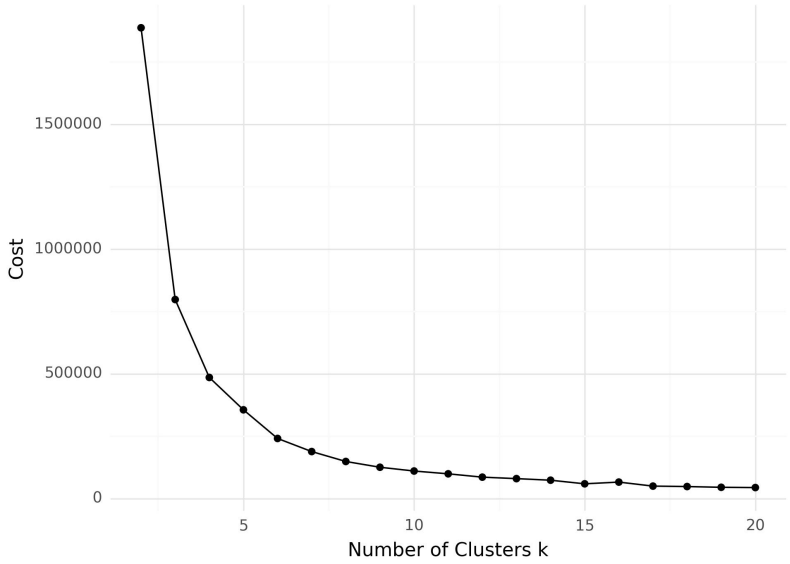


Figure 5.2: Example Image of Elbow plot used to decide numbers of clusters for optimum clustering results

When the value of k in the equation mentioned earlier is smaller than the actual clustering number, the decrease in J will be significant due to the substantial rise in the clustering degree of each cluster resulting from an increase in k. As k approaches the genuine clustering number, the clustering degree will decrease rapidly with increasing k. The decline in P will initially be significant, but it will gradually level off as P increases. Hence, the relationship graph between P and k exhibits an elbow shape, and the numerical value at this elbow is considered the actual clustering number of the data[47].

In the elbow graph example that was presented before, a range of clusters ranging from 1 to 12 was selected. The number of clusters is expressed along the X axis, and the cost or error is represented along the Y axis. Both axes are perpendicular to one another. The graph illustrates a slope of a decreasing cost function that increases with the number of clusters. After cluster 10, the slope of this graph does not decrease significantly, and the value of the cost does not decrease by a significant amount despite the increasing number

of clusters involved. Consequently, the number of clusters in the dataset should be a minimum of 10 for optimal results.

Silhouette Score

The purpose of the silhouette score is to assess the degree of separation and distinctiveness between clusters. It offers a means of evaluating the quality of clustering by presenting a measure of how closely an object corresponds to its own cluster in comparison to other clusters[48]. The silhouette plot displays the silhouette score for each iteration with K clusters. This score is calculated using separate silhouette coefficients for each data point and an average silhouette score for all clusters.

$$Silhouette(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{5.2}$$

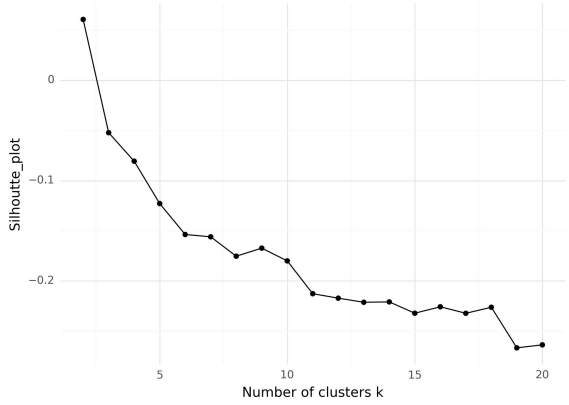


Figure 5.3: Example Image of Silhouette score plot used to decide numbers of clusters for optimum clustering results

Silhouette score ranges between -1 and 1. The value of the Silhouette score in negative indicates that object is poorly matched to the the cluster and have a similarity with the neighboring cluster. When the score is near 0, it indicates that the object is close to the decision boundary between two clusters. It suggests that the object is neither well-matched to its own cluster nor poorly matched to the neighbouring clusters[49]. This scenario can happen when data points in a region where it could belong to multiple cluster and it is ambiguous.

Davies-Bouldin Score plot

This matrix is used for the evaluation of compactness and separation between clusters. Davies-Bouldin score evaluates compactness and separation between Kclusters quantitatively. This score is calculated for each cluster pair and it is defined as the average similarity between each cluster and its most similar cluster. Intra-cluster and inter-cluster

similarities are used to compare to determine similarity for the Davies-Bouldin score.

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\frac{avg_i + avg_j}{d(c_i, c_k)} \tag{5.3}$$

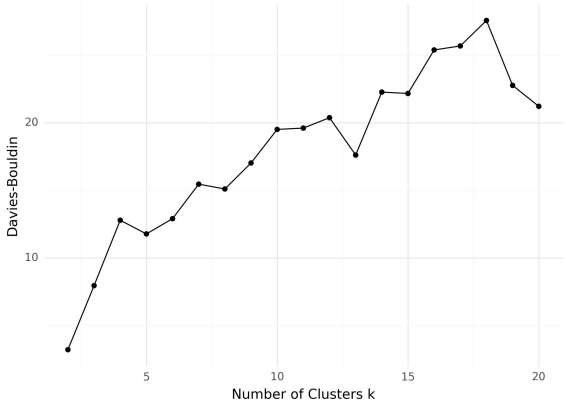


Figure 5.4: Example Image of Davies-Bouldin Score plot used to decide numbers of clusters for optimum clustering results

The Davies-Bouldin index ranges from 0 to ∞[50]. A higher value of the index for different numbers of clusters suggests poor clustering quality. It indicates less separation between clusters. A lower value for the Davies-Bouldin index is desirable which represents better clustering quality. The low index number shows that the average distance between clusters is large.

Calinski-Harabasz Index Plot

The Calinski-Harabasz Index is alternatively referred to as the variance ratio criteria. This indexing is used to determine the quality of clustering in the dataset. The Calinski-Harabasz index quantifies the ratio between the variance among different clusters and the variance within each cluster. The Calinski-Harabasz index is increased when there is a low variance across clusters and a high variance within clusters.

$$CH = \frac{B(k)}{(k-1)} \cdot \frac{N-k}{k-1} \tag{5.4}$$

There is no defined numerical range of the Calinski-Harabasz index[51]. The practical value of the index depends upon the data in the real world. When the variance between clusters is substantially greater than the variance within the cluster, the value of Calinski-Harabasz is higher. The clustering is more optimal when the Calinski-Harabasz index is high.

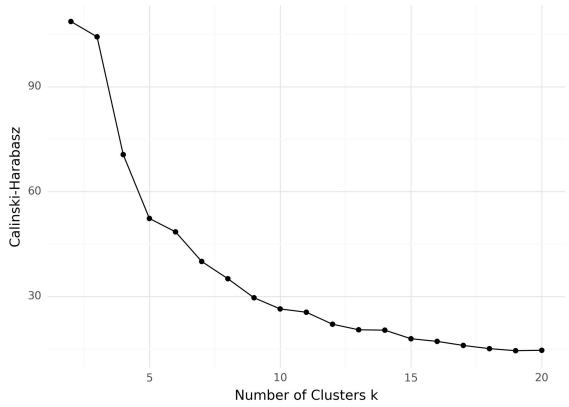


Figure 5.5: Example Image of Calinski-Harabasz Index plot used to decide numbers of clusters for optimum clustering results

5.3.2 Cluster Initialization

Cluster initialization, commonly referred to as cluster seeding, involves establishing the cluster centroids at the very first step of the iteration process. One major issue with KMeans and related algorithms like KModes and KPrototypes is that it chooses the initial seed at random from the sample space, which means it may be from the same space and isn't very diverse[52]. which can provide sub-optimal results. Therefore initialization of clustering methods are introduced.

Two primary initialization methods are commonly employed for KModes and KPrototype clustering. The related initialization methods are referred to as 'Huang' and 'Cao'.

Huang Initialization method

The Huang initialization approach utilizes a dissimilarity measure intended for categorical data. The selection process aims to minimize the total dissimilarity between data points and their respective cluster centroids. It aims to choose initial centroids that are representative of the different modes present in the data.

Cao Initialization method

The Cao initialization method is an alternative technique for initializing the cluster centroids in KModes clustering. Its purpose is to optimize the algorithm's speed of convergence.This method utilizes an approach that effectively balances the distance and density of instances. The centroids are chosen based on their dissimilarity to other data points and their density within the dataset[53].

The provided pseudocode 1 illustrates the fundamental steps of a clustering algorithm. Once the process is initiated, it begins by allocating cluster centroids K using the selected initialization method from 5.3.2. Subsequently, it undergoes an iterative loop, as demonstrated in the K-modes and K-prototypes. The objective of the algorithm is to min-

imize the error P. When the latest updated centroid achieves a value that is equal to or greater than the value in the previous iteration, it indicates that the process has reached the minimum point and has converged. The cluster performance is evaluated by metrics in 5.3.1. To find an optimum numbers of clusters, performance score with various numbers of clusters must be monitored. After that, these performance scores are compared and the optimum number of cluster is chosen.

Algorithm 1: Pseudo-code for Generic Clustering Algorithm for K-Means, K-Modes, and K-Prototypes

```
Input: Data, Number of clusters (k), Maximum iterations
Output: Final clusters, Final centroids
Initialization:
    Initialize centroids: centroids = initialize_centroids(data, k)
Training loop:
    for iter in range(max_iterations):
        Assign data points to clusters: clusters = assign_to_clusters(data,
  centroids)
        Update centroids: new_centroids = update_centroids(clusters)
        if convergence_criteria(centroids, new_centroids):
            break
        centroids = new_centroids
Finalization:
    Final clusters, final centroids = clusters, centroids
```

5.4 Application of Random Forest Algorithm

In this thesis work, this method is utilized to reveal hidden relations or degrees of dependency between each feature and the intended error class. This particular method makes use of a dataset that include characteristics from both the vehicle dataset and the charging session dataset. Nevertheless, a large dataset has been divided that takes into account each of the four main types of errors. For the purpose of preventing bias in a dataset, datapoints that are free of errors are chosen at random. However, these datapoints cover all charging behavior that is referred to by various clusters.

Since Random Forest is a type of Supervised Machine Learning algorithm, it requires training in order to accurately predict and produce output that closely matches the actual value. In this scenario, the algorithm must undergo training using the output class to determine if the error is true or false. Consequently, that results in a binary classifying output depending on the features included in the dataset.

Random Forest, unlike Neural Network and other supervised learning methods, has a more interpretable internal process and properties[54]. The decision making process in a Random Forest relies on a decision boundary and a set of rules in the nodes of each

Decision Tree, which may be easily comprehended. Thus Random Forest algorithms are considered as gray box model which can be interpreted and results behind particular output can be traced back using various methods[55].

5.4.1 Training of a Random Forest

The random forest algorithm is trained using a bagging also known as bootstrap aggregating technique in a machine learning. The process involves the random selection of subsets from the training data, training a model on these smaller selected datasets, and either averaging or keeping a mode the predictions as a final output[56].
Bootstrap samples are generated by randomly selecting N samples from the dataset. During the procedure, certain samples may be repeated while others may be omitted. These samples that have been generated are referred to as Bags. Bootstrapping is an essential initial stage in the training process of random forest.

Decision trees are created utilizing the aforementioned splitting procedures in the decision tree. The hyperparameters described below are utilized to regulate the quantity of trees and influence the structure and attributes of the trees.

To provide a final class prediction output the majority vote or mode of the outputs from individual trees are taken which is called Ensemble Aggregation. The accuracy of the prediction output of the random forest classifier is significantly influenced by hyperparameter adjustment. The following section provides a detailed discussion of hyperparameters for random forest classifiers.

5.4.2 Hyperparameters of Random Forest

Number of Trees
The number of trees is a crucial hyperparameter that must be optimized in a random forest model. The selection of a number of trees in a random forest is generally a trade-off between computational cost and performance. Generally, adding more trees in the random forest increases the accuracy of the model. On the other hand, it increases computational resources required[57].

Tree Depth
This hyperparameter is alternatively referred to as the maximum depth. The tree depth parameter specifies the upper limit on the number of levels or depth that a decision tree can have in a machine-learning model. The depth of a tree is determined by the length

of the longest path from the root node to a leaf node. A lower tree depth can result in underfitting, whereas a deeper tree depth might lead to overfitting[58].

Number of Features

The number of features, or the maximum number of features, in a random forest specifies the maximum number of features that are considered to split a node in each decision tree of the forest. During the creation of an individual tree, it maintains the randomness level in feature selection. By incorporating this hyperparameter, the occurrence of overfitting is reduced. Typically, two strategies are employed to determine the number of attributes to be selected. The following strategies are: 1. Establishing a limit on the number of features to be equal to the square root of the total number of characteristics. 2. Establishing a limit on the number of features based on the Base-2 logarithm of the total number of features.

Minimum Samples per Leaf

During the training phase, the decision tree grows by generally splitting nodes based on features, creating a hierarchical structure. This hyperparameter is utilized to determine the minimum number of samples necessary to form a leaf node while constructing a decision tree in a random forest. This hyperparameter restricts the minimum number of samples that must be present in the leaf node. Insufficient minimum sample numbers at the leaf node result in poor generalization and a tendency to overfit.

Algorithm 2: Pseudo-code for Random Forest

```
1  Define Hyperparameters:
2      number_of_trees, max_depth, max_features
3  Training and Evaluation:
4      train_data, test_data, train_labels, test_labels = train_test_split(data,
       labels, test_size, random_state)
5      rf_classifier = train_random_forest_classification(train_data, train_labels,
       num_trees, max_depth, max_features)
6      test_accuracy = evaluate_random_forest_classification(rf_classifier,
       test_data, test_labels)
7  Functions:
8      Function train_random_forest_classification(data, labels, num_trees,
       max_depth, max_features):
9          Train Random Forest Classifier with specified hyperparameters
10         Return: trained_model
11     Function evaluate_random_forest_classification(model, test_data,
       test_labels):
12         Evaluate Random Forest Classifier on test data
13         Return: accuracy
```

Internal Validation during Training

The Out-of-the-bag (OOB) score in the random forest technique allows for convenient validation of the algorithm during training, eliminating the need for a separate validation dataset.[59] During the bagging process of a training dataset with N samples some samples are not utilized in the training process of the particular tree which are called Out-of-the-bag samples.

Interpretation of the Random Forest Model

There are several Global Model-Agnostic Methods that are not dependent on the specific nature of the model. These methods are employed to analyze the overall model behavior of the machine learning algorithm. These systematic approaches try to provide broad insights and explanations on how the algorithm predicts in the dataset. This finally improves the comprehensibility and clarity of the model. Instead of providing insights based on individual predictions, these methods analyze the entire dataset. One of the widely used agnostic methods is Permutation Feature Importance.

5.4.3 Permutation Importance of Features in Random Forest

Permutation importance calculates the feature importance of estimators for a dataset. This model inspection method is used for any machine learning model where data is tabular. This method involves recursively permuting features for a specified number of iterations. The values of a selected feature are shuffled in the method, while the values of other features remain unchanged, and the previously trained random forest algorithm is then applied to this tempered dataset. Next, the predicted output of this modified dataset is taken. When the classification approach is employed, the class labels are predicted using the adjusted dataset. After applying the same method to the remaining features, variances in output predictions are determined after this method has been applied. When there is a significant change in the label that is predicted after a certain feature has been shuffled, then that particular feature is considered to be the important feature in the dataset that has been provided. Nevertheless, the usefulness of this approach depends heavily on the accuracy of the trained machine learning model. Suppose the model is inadequately trained and fails to produce accurate predictions. In that case, there is a possibility that the permutation importance method will not accurately identify the features that significantly affect the class prediction. The permutation importance method can be used on both test and training samples. But when the data that the model was trained on are used, the results are usually too optimistic. That means the model is more accurate when training data is used. If the model is overfitting in some way, it won't be able to accurately find actual important features. So, a test dataset or unseen dataset is used to find important features in a dataset.[60]

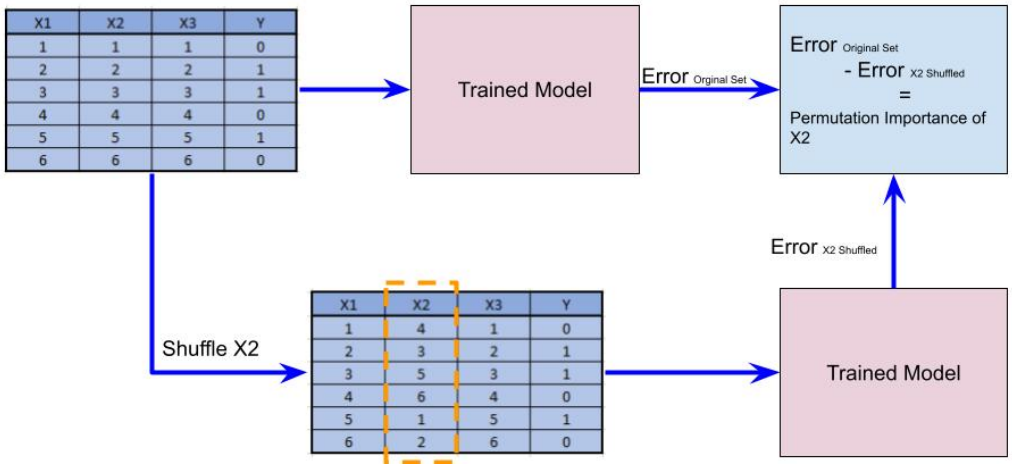


Figure 5.6: Permutation Feature Importance calculation of feature X2

One of the most prominent features in the dataset may overwhelm other correlated features, which is the primary obstacle of utilizing this method of assessing model behavior. In this situation random forest uses only this highly correlated feature in order to provide a prediction. This problem occurs when there are several collinear features present in the dataset.

An iterative approach can be used to identify additional underlying correlated features. When a prominent feature is identified that is overshadowing other significant correlated features, it is removed in the second iteration and a consequent drop in model accuracy is observed. This iterative process is continued until the random forest generates predictions based on multiple combinations rather than solely depending on a single dominant characteristic. This methodology is tedious, time-consuming and computationally expensive for datasets with large numbers of features.

A hierarchical clustering of features has been carried out in order to reduce the number of iterations of the random forest method and to improve the efficiency of the process. It is Through using the method of hierarchical clustering, the expected outcomes of hierarchical clusters of colinear features of the dataset. The hierarchical clustering approach has proven to be effective in determining whether or not the ordinal features of the dataset show colinearity or correlation. Spearman's rank order correlation has been applied to determine the presence of correlation. The Spearman's correlation coefficient is a non-parametric measure of the monotonicity of the relationship of variables. The value of the Spearman's correlation coefficient varies between -1 to +1. Where 0 means no correlation between variables. The positive value of the coefficient signifies the positive correlation of variables when one increases and then there is an increasing change in other variables as well. While the negative coefficient value means the negative correlation of variables.

The method to calculate Spearman correlation,

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{5.5}$$

in the expression, ρ is the Spearman rank correlation coefficient. n is the number of paired observations or data points in the sample. and d_i represents the difference between the ranks of paired observations.
After computing a Spearman correlation coefficient of variables, values are converted into a Dissimilarity matrix. Because the Dissimilarity measure is more suitable and can be used for clustering methods. Dissimilarity matrix D_{ij} are calculated using,

$$D_{ij} = 1 - |\rho_{ij}| \tag{5.6}$$

The absolute value ensures that the dissimilarity values are always non-negative, and subtracting it from 1 transforms the correlation values into dissimilarity values.

Using the Hierarchical clustering method, a Dendrogram has been created. On the X-axis of the dendrograms, features of the dataset are placed as dendrogram leaves. These features connect to the most correlated neighboring feature. Then the threshold of the Dendrogram is manually picked to group these features into clusters. One feature from each cluster is selected and from these selected features new random forest is trained. The random forest method has a decrease in accuracy when the number of features are lowered. However, if the test accuracy surpasses a significant threshold and the algorithm accurately predicts the output labels, then the specified threshold for the random forest feature is satisfactory. Decreasing the threshold leads to a surge in the number of hierarchical clusters, while the number of features for random forest algorithm also grows. After the Random Forest algorithm has been trained, the feature permutation importance score for each of the features that were chosen has been recalculated. The importance of each of the features that fall within that cluster is represented by the permutation score for these particular features.

Chapter 6

Results Discussion

This chapter delves into the evaluation and observations of applied algorithms, notably Random Forest and Clustering algorithms. Through meticulous analysis, their performance and effectiveness across various scenarios are scrutinized, providing valuable insights into their applicability and potential enhancements. This comprehensive examination contributes to advancing understanding and utilization of algorithmic methodologies in real-world contexts.

6.1 Evaluation of Clustering

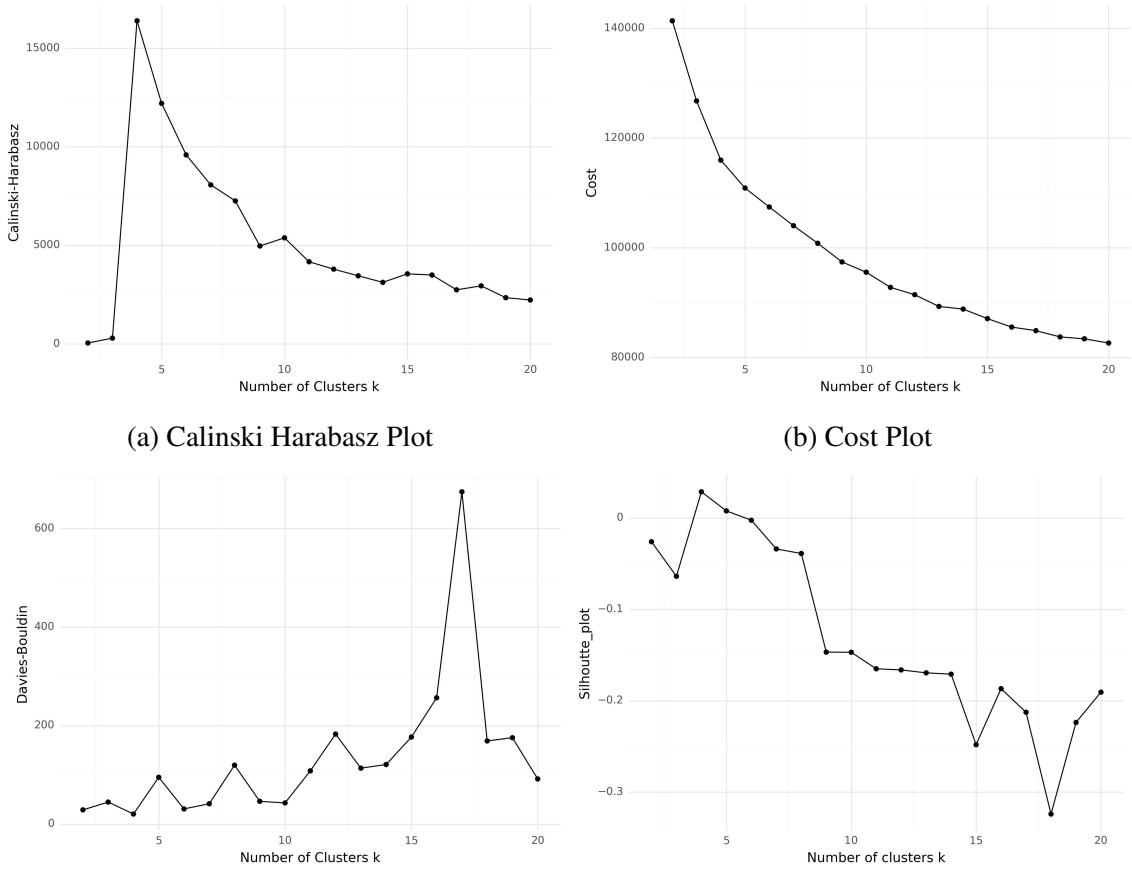
Various clustering algorithms, including K-means, K-mode, and K-prototype, were evaluated, with the K-prototype algorithm being chosen due to its ability to handle diverse data types in the dataset effectively. The selection of an optimal number of clusters (K=8) for the K-prototype clustering was based on several evaluation scores discussed in the "Application of Clustering" section. The decision was informed by observations from graphs presented in Table 6.1.

Upon closer examination of these graphs, the Calinski-Harabasz plot 6.1a showed a consistent variance ratio across increasing numbers of clusters, suggesting a plateau effect. The Cost plot or elbow plot 6.1b did not exhibit a clear elbow point but rather a more gradual slope between 5 to 10 clusters. Although cost plots are commonly used for determining the number of clusters, this gradual slope may be due to overlapping data points.

The Davies-Bouldin plot 6.1c showed a significant increase in the value on the X-axis, with the desired value being close to 0. A good Silhouette score 6.1d, ranging from 1 to -1 and closer to 0, was also considered favorable for cluster analysis.

These evaluation factors collectively influenced the choice of K=8 for the cluster analysis. Furthermore, noise in the data was identified during clustering analysis, leading to the normalization of numerical data before generating the heatmap for further insights into

data patterns.Despite the challenge of achieving distinct clustering with the K-Prototypes algorithm, it was ultimately chosen for its superior performance over alternative algorithms. While the data may not have clustered as distinctly as desired, the algorithm demonstrated effectiveness in capturing underlying patterns and relationships within the dataset. Therefore, despite the difficulty in analyzing certain clusters individually, the overall performance and ability to uncover valuable insights justify the selection of the K-Prototypes algorithm for the analysis. The heatmap referenced in Table 6.1 is depicted in Figure 7.2. (The heatmap is provided in the appendix due to its unusual size.)



(a) Calinski Harabasz Plot

(b) Cost Plot

(c) Davies Bouldin Plot

(d) Silhouette Plot

Figure 6.1: Plots for K-Prototype clustering to select an optimum number of clusters

Cluster	Observation
1	Cluster 1, constituting approximately 35% of the entire dataset, is the largest among all clusters. However, deriving useful information from this large cluster is challenging due to its diversity. The cluster encompasses data with and without errors and even other features are not clearly distinguishable making it difficult to derive specific observations.
2	Cluster 2, predominantly consists of vehicle charging data from the USA and Canada regions. A distinct separation is evident in the heatmap regarding latitude, longitude, and country name, with data from Europe present in other clusters. Charging data associated with CPOs Electrify America and Electrify Canada, utilizing the OCPI protocol, is logged within this cluster. Additionally, it's observed that in the USA and Canada, vehicles with larger battery configurations are more prevalent. Charging in this cluster primarily occurs with 150KW, 350 kW and then 50KW power.
3	In cluster 3, some portion of the heatmap show that during the charging process, vehicle has charged slowely and the temperature of the BMS is low. which shows the majority of slow AC charging is used. Thi cluster has also the lowest average of the amount of power used to maintain thermal requirements of the battery.
4	In cluster 4, data with lowest numbers of errors are logged. This cluster can be seen as a reference point where vehicles undergo the standard charging process. However, it also includes some data points where errors occurred, but these can be disregarded for the sake of overall representation and analysis.
5 & 6	Most LED notification errors cluster mainly in clusters 5 and 6. Data points where errors were detected Cluster 2 predominantly consists of vehicle charging data from the USA and Canada regions. A distinct separation is evident in the heatmap regarding latitude, longitude, and country name, with data from Europe present in other clusters. Charging data associated with CPOs Electrify America and Electrify Canada, utilizing the OCPI protocol, is logged within this cluster. Additionally, it's observed that in the USA and Canada, vehicles with larger battery configurations are more prevalent. Charging in this cluster primarily occurs with 150 kW and 350 kW power.but charging continued also fall into these clusters. Additionally, power-related metrics including maximum DC power, current, and BMS temperature are notably low in these clusters. The primary distinction between clusters 5 and 6 lies in the Start SOC of the BMS. Cluster 5 exhibits a higher Start SOC, whereas cluster 6 consistently shows a lower Start SOC when comparing these two clusters. In these clusters, vehicles are primarily charged at charging stations like Ionity, Compleo, Charge Point Austria, and Allego. In these clusters, vehicles are mainly charged using high-power DC charging, including 350 and 150 kW chargers. Additionally, some vehicles in this cluster are charged using 22 kW chargers.Upon closer observation, it is noted that the charging electronics versions distinguished in clusters 4, 5, and 6 exhibit more errors compared to other clusters.
7 & 8	In clusters 7 and 8, vehicles with low SOC during the plug-in process are clustered, but no specific pattern or anomaly is evident.

Table 6.1: Observations of Clusters derived from K-Prototypes Clustering

6.2 Evaluation of Random Forest

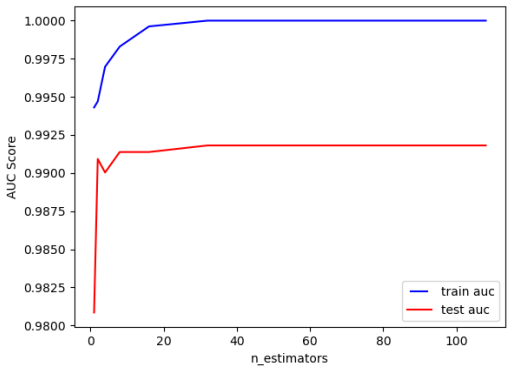
To prevent bias in the predicted outcome, different datasets are created for each error type. Subsequently, the datasets are divided into an 80:20 ratio for the purpose of training and testing the random forest model. The optimal number of estimators is determined by iteratively training Random Forest models with different numbers of estimators. The performance of each model is evaluated using the ROC-AUC metric on both the training and test sets. ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve) are metrics used for evaluating the accuracy of classification models. This method has been widely used in the evaluation of binary classification tasks.

The trade-off between the true positive rate and the false positive rate is represented graphically by the ROC curve. The percentage of actual positive instances that the model has accurately identified is known as the true positive rate. While the false positive rate is the proportion of actual negative instances incorrectly predicted as positive by the model. ROC curve is then taken for all determined numbers of estimators. the ROC curve summarises the confusion matrices for each iteration produced.

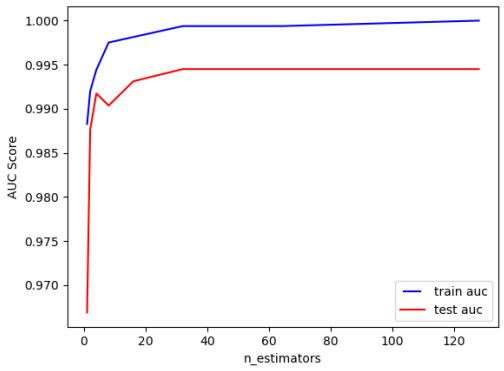
AUC is the quantification of the overall performance of a classification. It is commonly used to compare different models. Here the area under the curve of ROC plot generated by iteration with varying numbers of estimators is plotted.

A model with a higher ROC-AUC curve generally indicates better performance. A diagonal random guessing on the ROC plot has an AUC of 0.5 and a perfect model would have an AUC of 1.0. However, AUC 1.0 is ideal but it generally indicates instead of generalizing the pattern the model is overfitted.

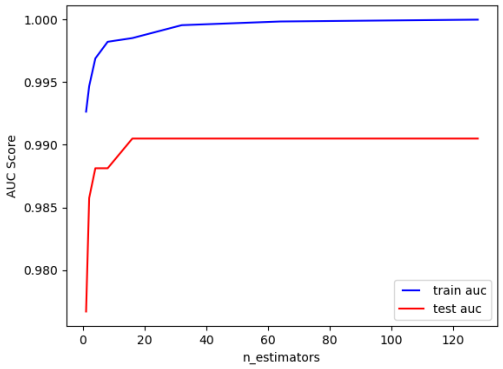
6.2 is a group of figures where ROC-AUC graph of train and test runs of the random forest algorithm of all different errors. 6.2a illustrates that the model starts showing overfitting of the data after 25 estimators in the random forest. During test runs, it has been observed that the random forest method has the highest performance for the prediction when using 22 decision trees. Referring to the graph the training AUC score is 0.997 and the test AUC score is 0.990. 6.2b illustrates that the model starts showing overfitting of the data after 32 estimators in the random forest during test runs, it has been observed that the random forest method has the highest performance for the prediction when using 32 decision trees. Referring the the graph the training AUC score is 0.997 and the test AUC score is 0.991. 6.2c also illustrates the training AUC score of 0.998 with 32 estimators. Increasing numbers of estimators further led to overfitting of the model. The test AUC score for the third graph is 0.991.



(a) Error Detected but Charging Continued



(b) Error Detected and charging stopped



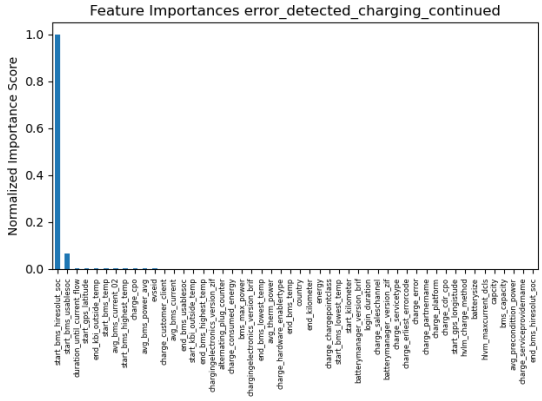
(c) LED error notification

Figure 6.2: ROC-AUC curve of Errors

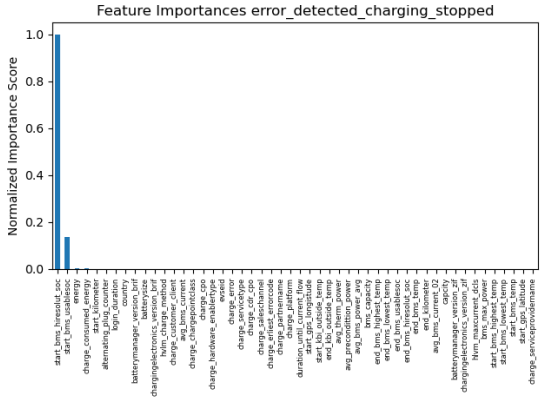
These estimators provide effective predictions of the target output on the test data and provides the important features using permutation combination method. However, as mentioned in the preceding section 5.4.3, Random Forest models may choose to utilize only a small subset of the most significant or highly correlated features, so concealing the importance of other relevant features. This result is evident in sub-figures of 6.3. In these graphs only 1 or 2 features are highlighted as an important features. In the following sections results of important features using the permutation importance method and important features using permutation importance using only non-correlated features are described.

6.2.1 Hierarchical clusters of features of Errors

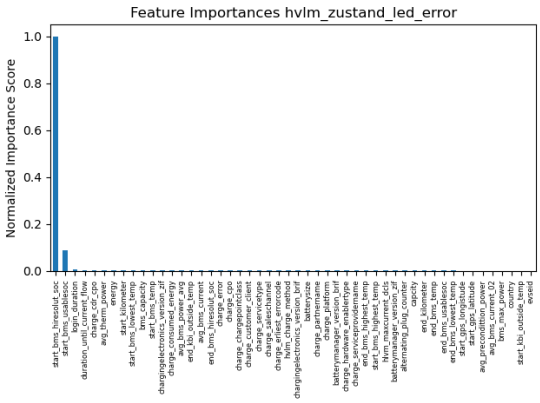
In this section, hierarchical clusters are derived to visualize the non-collinearity of features clusters for permutation feature importance. It is important to note that with increased distance in the hierarchical clustering, the collinearity or similarity of features reduces. The specific errors are detailed on the X-axis of the Dendrogram, while the distances between features are depicted on the Y-axis. Distinct sets of hierarchical clusters are denoted by different colors in binary dendrogram trees. Figures 6.4, 6.5, and 6.6 illustrate the dendrograms of errors.



(a) Error detected but charging continued



(b) Error Detected and charging stopped



(c) LED error notification

Figure 6.3: Feature Importance graph using all features in the random forest

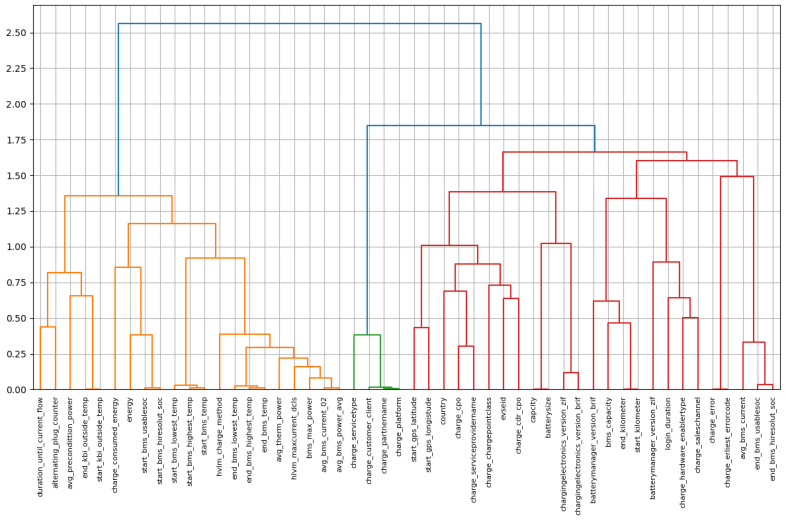


Figure 6.4: Dendrogram of features of error - Error detected but charging continued

6.2.2 Selection of Threshold distance to cut Dendrogram

An acceptable threshold or cut distance is selected during the creation of hierarchical clusters, as elaborated in 6.2.1. Trimming the dendrogram becomes necessary to choose non-correlated features from the dataset. Visual inspection of dendrograms is employed to determine the optimal cutting distance, resulting in the selection of two distances for

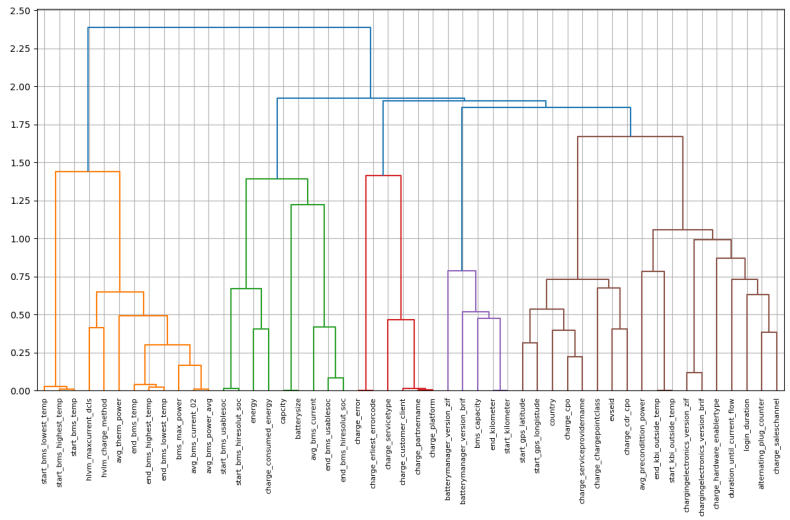


Figure 6.5: Dendrogram of features of error - Error detected and charigng stopped

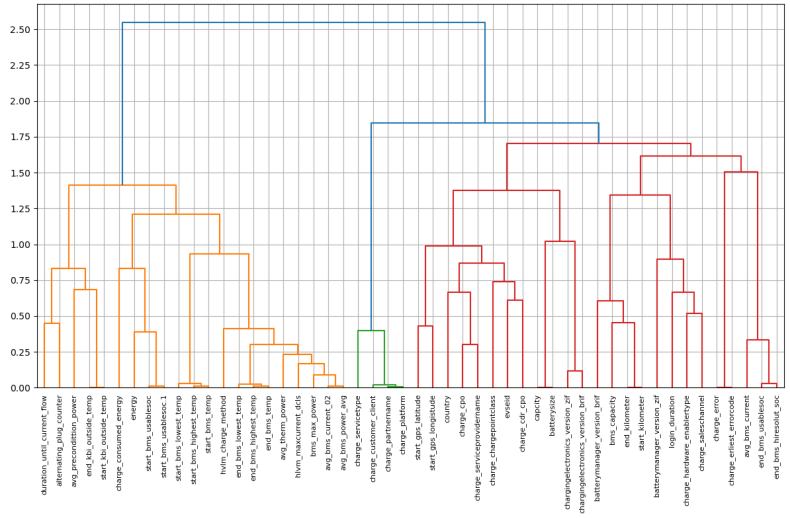


Figure 6.6: Dendrogram of features of error - LED notification Error

each dendrogram. These features are subsequently utilized to train and test the random forest algorithm. The tables 6.2, 6.3, and 6.4 present the test accuracy at various threshold distances. As threshold distance is increased, the number of hierarchical clusters decreases. As discussed previously, only one feature per cluster is selected to achieve a higher degree of noncollinearity among features. It is evident that with increasing distance, the test accuracy of the random forest model decreases. This trade-off between feature noncollinearity and model accuracy must be carefully considered during the selection of threshold distances for hierarchical clustering. However as Spearman's rank order only computes ordinal data efficiently, nominal features are still considered separately in order to avoid loss of information.

Threshold Distance	Test Accuracy
0.80	0.92
1.01	0.86

Table 6.2: Dendrogram threshold distance of Error Detected but Charging continued

Threshold Distance	Test Accuracy
1.01	0.79
1.50	0.73

Table 6.3: Dendrogram threshold distance of Error Detected and charging stopped

Threshold Distance	Test Accuracy
0.80	0.92
1.02	0.90

Table 6.4: Dendrogram threshold distance of LED Notification Error

6.2.3 Important features for Error detected but charging continued

In this section, the important features for the "Error detected but charging continued" are discussed. The feature importance score is normalized between 0 and 1. In the graph 6.7, the dendrogram is cut at 0.8 to visualize the important features while in the graph 6.8 dendrogram is cut at 1.01. In these graphs, important features are illustrated in descending order from left to right, with the most influencing feature on the outcome mentioned on the leftmost bar of the graph.

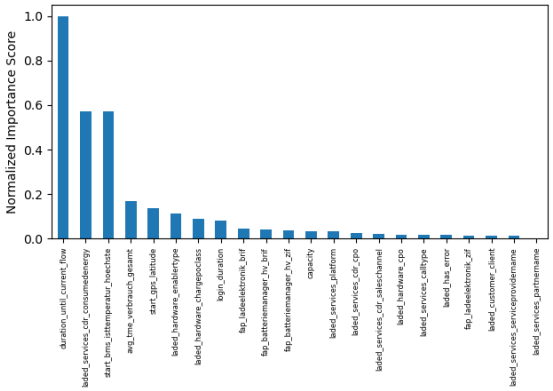


Figure 6.7: Noncollinear feature importance of Error detected but charging continued at 0.8 Threshold

6.2.4 Important features for Error detected and charging stopped

The feature importance graphs in this section are similar to those explained in 6.2.3. In 6.9, important features for Error detected and charging stopped are displayed, which were derived using a threshold of 1.01. Similarly, in 6.10, important features derived at a threshold of 1.50 are illustrated. Comparing the feature importance graphs at different

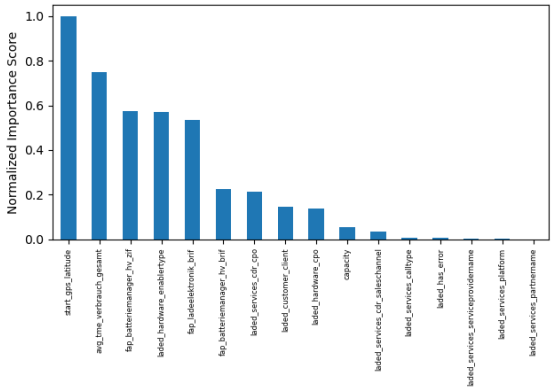


Figure 6.8: Noncollinear feature importance of Error detected but charging continued at 1.01 Threshold

	Most influencing feature groups when Dendrogram 6.4 threshold is at 0.8 6.7
1	The most influential feature group revolves around 'duration_until_current_flow', encompassing all features up to 'start_kbi_outsideß$_temp'$.
2	The second most influential feature group is related to 'laded_services_cdr_consumed_energy', including features up to 'start_bms_hiresolutsolute'.
3	The third most influential cluster group is associated with 'start_bms_highest_temp', which includes features such as 'start_bms_lowest_temp' and 'start_bms_temp'.

Table 6.5: Most influencing feature groups for Error detected but charging continued at 0.8 threshold

	Most influencing feature groups when Dendrogram 6.4 threshold is at 1.01 6.8
1	The most influential feature group revolves around 'start_gps_latitude', encompassing all features up to 'charge_cdr_cpo'
2	The second most influential feature group is related to 'avg_tme_verbrauch_gesamt', including features up from 'duration_until_current_flow' to 'avg_xbms_power_avg'.
3	The third most influential cluster group is associated with 'fap_battery_manager_brif', which includes features up to 'start_kilometer'.

Table 6.6: Most influencing feature groups for Error detected but charging continued at 1.01 threshold

thresholds provides insights into how the importance of features changes with varying levels of clustering.

	Most influencing feature groups when Dendrogram 6.5 threshold is at 1.01 6.9
1	The most influential feature group revolves around 'start_bms_highest_temp', which includes features such as 'start_bms_lowest_temp' and 'start_bms_temp'
2	The second most influential feature group is related to 'start_gps_latitude', including features up to 'charge_cdr_cpo. Note This group of cluster includes features which are not nominal but ordinal 6.2.2
3	The third most influential cluster group is associated with 'avg_precondition_power', which includes features such as 'end_kbi_outside_temp' and 'start_kbi_outside_temp'.

Table 6.7: Most influencing feature groups for Error detected and charging stopped at 1.01 threshold

	Most influencing feature groups when Dendrogram 6.5 threshold is at 1.50 6.10
1	The most influential feature group revolves around 'start_gps_latitude', encompassing all features up to 'charge'
2	The second most influential feature group is related to 'start_gps_latitude', including features up to 'charge_cdr_cpo'. Note This group of cluster includes features which are not nominal but ordinal 6.2.2
3	The third most influential cluster group is associated with 'avg_precondition_power', which includes features such as 'end_kbi_outside_temp' and 'start_kbi_outside_temp'.

Table 6.8: Most influencing feature groups for Error detected and charging stopped at 1.50 threshold

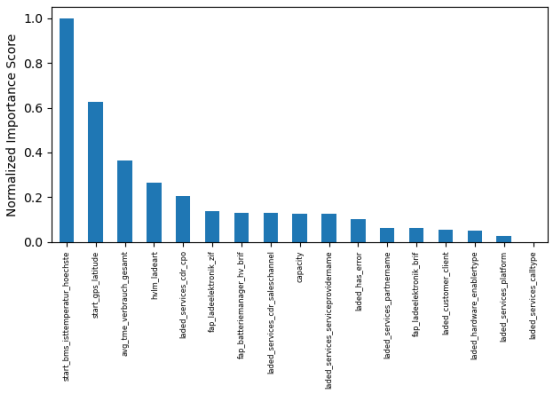


Figure 6.9: Noncollinear feature importance of Error detected and charging stopped at 1.01 Threshold

6.2.5 Important features for LED Notification error

In this section, graphs are generated in a manner similar to that explained in 6.2.3. In the feature importance graph 6.11, the dendrogram was cut at a threshold of 0.80. Conversely, in 6.12, the dendrogram was cut at a threshold of 1.08. These different thresholds provide

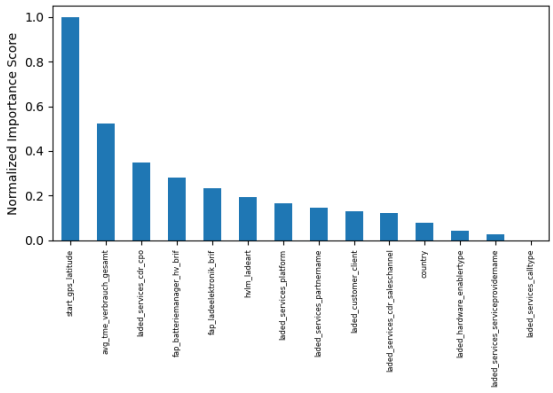


Figure 6.10: Noncollinear feature importance of Error detected and charging stopped at 1.50 Threshold

insights into how varying levels of clustering affect the importance of features in the context of the LED notification error.

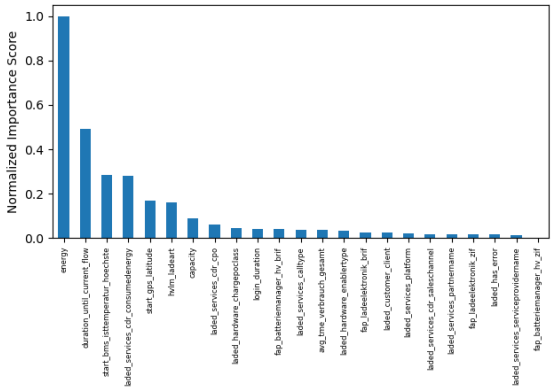


Figure 6.11: Noncollinear feature importance LED notification error at 0.80 Threshold

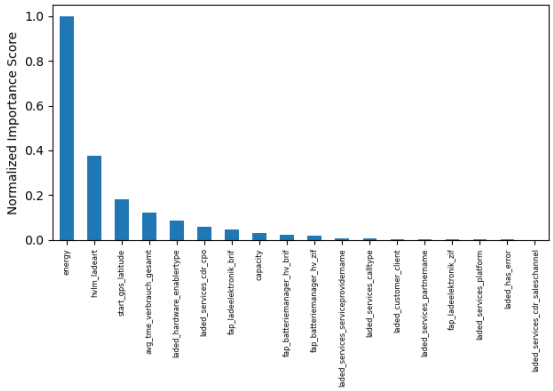


Figure 6.12: Noncollinear feature importance LED notification error at 1.08 Threshold

6.3 Evaluation of Principal Component Analysis

Principal Component Analysis (PCA) was conducted to validate the interdependencies or correlations of features identified in the hierarchical plots 6.4, 6.5, and 6.6 using the

	Most influencing feature groups when Dendrogram 6.6 threshold is at 0.80 6.11
1	The most influential feature group revolves around 'energy', encompassing features such as 'charge_consumed_energy' and 'start_bms_usablesoc'
2	The second most influential feature group is related to 'hvlm_charge_method', including features up to 'avg_bms_current_02'.
3	The third most influential cluster group is associated with 'start_gps_latitude' and 'start_gps_longitude'.

Table 6.9: Most influencing feature groups for LED Notification Error at 0.80 threshold

	Most influencing feature groups when Dendrogram 6.6 threshold is at 1.08 6.12
1	The most influential feature group revolves around 'energy', encompassing features such as 'charge_consumed_energy' and 'start_bms_usablesoc'
2	The second most influential feature group is related to 'hvlm_charge_method', including features from 'start_bms_lowest_temperature' to 'avg_bms_current_02'.
3	The third most influential cluster group is associated with 'start_gps_latitude' and features until 'charge_cdr_cpo'. Note This group of cluster includes features which are not nominal but ordinal 6.2.2

Table 6.10: Most influencing feature groups for LED Notification Error at 1.08 threshold

Spearman rank formula. In Figure 6.13, four principal components are derived. Each component illustrates the correlation between features, indicated by bars of four different colors. Each feature is represented by four bars of different colors in Figure 6.13. These bars are compared with bars representing other features to validate the correlation between them. A higher bar indicates a stronger correlation, while bars on the negative axis represent inverse correlations with the positive ones.

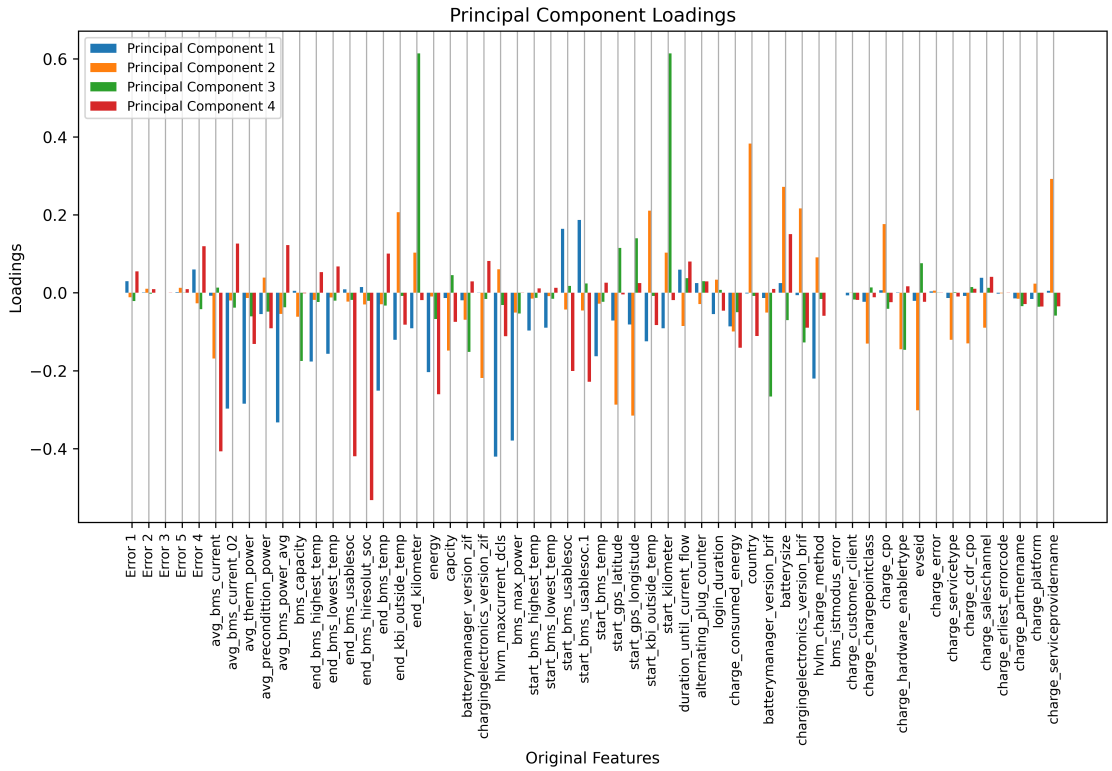


Figure 6.13: Principal Component Analysis of features

Chapter 7

Summary and Outlook

7.1 Summary

The primary objective of this thesis is to uncover underlying patterns within recorded charging event sessions. This involves conducting both vertical and horizontal analyses of the 2D dataset stored in the Database of the MSP.

The vertical analysis of the dataframe aims to uncover the interdependence among Key Performance Indicators (KPIs) monitored and logged during the charging process. Additionally, it investigates the extent of influence that these KPIs, mentioned as features in various sections of the thesis, have on charging errors.To conduct vertical analysis, a Random Forest algorithm was trained to predict error occurrences. The decision-making process of the Random Forest algorithm is traceable, enabling the identification of important features that indicate the possibility of an error. Initial results revealed that the algorithm heavily relied on one or two highly correlated features, which obscured other relevant features. To promote feature diversity and identify correlated features, a Hierarchical clustering method was employed. Spearman's ranking method was used to derive distance scores between features, aiding in the clustering process. This approach aimed to uncover a more comprehensive set of features contributing to error prediction and improve the overall analysis accuracy. This approach produces tabular results. This method analyses each error using two sets of relevant feature groups obtained from two separate dendrogram thresholds. Each set is divided into three important sections, with the top mentioned features being the most crucial.These feature groups are in 7.3, 7.4, 7.5.

The Horizontal analysis of the dataframe aims to reveal patterns or anomaly in the charging behaviour present in the dataset. This horizontal analysis also reveals trends in the dataset. For horizontal analysis of the dataframe, the K-Prototypes clustering algorithm was utilized. This algorithm clusters all recorded charging sessions in the dataset based on both numerical and categorical features. The optimal number of clusters was determined using evaluation methods such as Calinski-Harabasz, Cost plot, and others. The result-

ing clustered data was then visualized using a heatmap, and insights from the heatmap were recorded for analysis and interpretation. This approach facilitated the exploration of patterns and relationships among different clusters of charging sessions in the dataset.

7.2 Future Scope

The identification of influential features by the Random Forest algorithm in predicting errors, as demonstrated in this thesis, opens up avenues for future research. One potential extension of this work involves determining threshold values for these features, beyond which there is a higher probability of errors occurring. This exploration can provide valuable insights into preemptive measures or predictive models for error mitigation during charging events.

The clustering algorithm has grouped data based on features related to power and charge consumption. Leveraging this clustering method, it becomes possible to associate the charging behavior of newly incoming data into the database. This association facilitates easier pattern analysis of charging types and errors, enhancing the understanding and management of charging processes in the future. The association model can be integrated into the data pipeline, enabling it to assign association labels when new data is stored in the database. This automated process ensures that each new data entry is tagged with relevant cluster or association labels based on its characteristics related to charging behavior. Such implementation streamlines data management and facilitates ongoing analysis and decision-making processes based on the charging type and potential errors associated with each data entry.

7.3 Challenges During the Thesis

In the initial phase of the thesis, a sample dataset was provided by the company, allowing for training and testing of various algorithms. However, upon requesting the complete dataset after finalizing the algorithm and direction, a challenge arose. The vehicle manufacturer had altered pipelines and data sampling methodologies, resulting in differences in how KPIs were logged in the database. The new dataset exhibited significant quality issues, necessitating substantial time investment in waiting for the new data, understanding the new KPIs, preprocessing the data, and adapting algorithms accordingly.

References

[1] Why the automotive future is electric, Sep 2021.

[2] Global ev sales for 2022.

[3] Iea. Trends in charging infrastructure – global ev outlook 2023 – analysis.

[4] Yusheng Zhang. Analysis of battery swapping technology for electric vehicles–using nio's battery swapping technology as an example. In *SHS Web of Conferences*, volume 144, page 02015. EDP Sciences, 2022.

[5] Taylor M Fisher, Kathleen Blair Farley, Yabiao Gao, Hua Bai, and Zion Tsz Ho Tse. Electric vehicle wireless charging technology: a state-of-the-art review of magnetic coupling systems. *Wireless Power Transfer*, 1(2):87–96, 2014.

[6] Chirag Panchal, Sascha Stegen, and Junwei Lu. Review of static and dynamic wireless electric vehicle charging system. *Engineering science and technology, an international journal*, 21(5):922–937, 2018.

[7] Muhammad Shahid Mastoi, Shenxian Zhuang, Hafiz Mudassir Munir, Malik Haris, Mannan Hassan, Muhammad Usman, Syed Sabir Hussain Bukhari, and Jong-Suk Ro. An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends. *Energy Reports*, 8:11504–11529, 2022.

[8] E-mobility: plug type overview hella provides a brief overview of the most common plug and charging cable types, Jun 2023.

[9] General information on recharging systems.

[10] Lukas Schriewer and Jozsef Farkas. Importance of interoperability for a seamless ev charging experience.

[11] Joachim Globisch, Patrick Plötz, Elisabeth Dütschke, and Martin Wietschel. Consumer preferences for public charging infrastructure for electric vehicles. *Transport Policy*, 81:54–63, 2019.

[12] Mart van der Kam and Rudi NA Bekkers. Comparative analysis of standardized protocols for ev roaming: Report d6. 1 for the evroaming4eu project. 2020.

[13] Open chargepoint protocol 2.0.1 introduction. 2020.

[14] Dwidharma Priyasta, Hadiyanto Hadiyanto, and Reza Septiawan. An overview of ev roaming protocols. In *E3S Web of Conferences*, volume 359, page 05006. EDP Sciences, 2022.

[15] International Electrotechincal Commission. Iec 63110-1:2022 protocol for management of electric vehicles charging and discharging infrastructures - part 1: Basic definitions, use cases and architectures. page 314, 2022.

[16] Intercharge enables europe-wide charging of electric vehicles 2013. 2013.

[17] Ocpi 2.2.1: Open charge point interface - evroaming foundation. 2021.

[18] Iso/iec 2382-14:1997(en) information technology — vocabulary — part 14: Reliability, maintainability and availability.

[19] Mattias Nyberg. *Model based fault diagnosis: Methods, theory, and automotive engine applications*. PhD thesis, Linköping University, 1999.

[20] Qingsong Yang. *Model-based and data driven fault diagnosis methods with applications to process monitoring*. Case Western Reserve University, 2004.

[21] Brian S Lindner and Lidia Auret. Data-driven fault detection with process topology for fault identification. *IFAC Proceedings Volumes*, 47(3):8903–8908, 2014.

[22] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques third edition. *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*, 2012.

[23] Ralf Hartmut Güting. An introduction to spatial database systems. *the VLDB Journal*, 3:357–399, 1994.

[24] Li M Chen, Zhixun Su, Bo Jiang, and Li M Chen. Machine learning for data science: Mathematical or computational. *Mathematical Problems in Data Science: Theoretical and Practical Methods*, pages 63–74, 2015.

[25] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

[26] Qiong Liu and Ying Wu. Supervised learning. *Encyclopedia of the Sciences of Learning*, page 3243–3245, 2012.

[27] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[28] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[29] R Suwanda, Zulfahmi Syahputra, and Elvi M Zamzami. Analysis of euclidean distance and manhattan distance in the k-means algorithm for variations number of centroid k. In *Journal of Physics: Conference Series*, volume 1566, page 012058. IOP Publishing, 2020.

[30] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[31] Taeho Jo. *Machine learning foundations*. 2021.

[32] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[33] Paul E Utgoff, Neil C Berkman, and Jeffery A Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29:5–44, 1997.

[34] Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali. A comparative study of decision tree id3 and c4. 5. *International Journal of Advanced Computer Science and Applications*, 4(2):13–19, 2014.

[35] Bhumika Gupta, Aditya Rawat, Akshay Jain, Arpit Arora, and Naresh Dhami. Analysis of various decision tree algorithms for classification in data mining. *International Journal of Computer Applications*, 163(8):15–19, 2017.

[36] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[37] Roger J Lewis. An introduction to classification and regression tree (cart) analysis. In *Annual meeting of the society for academic emergency medicine in San Francisco, California*, volume 14. Citeseer, 2000.

[38] Shan Suthaharan and Shan Suthaharan. Decision tree learning. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, pages 252 – 255, 2016.

[39] Shan Suthaharan and Shan Suthaharan. Decision tree learning. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, pages 250 – 252, 2016.

[40] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[41] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van

Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] Nelis J. de Vos. kmodes categorical clustering library. https://github.com/nicodv/kmodes, 2015–2021.

[44] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[45] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[46] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[47] Fan Liu and Yong Deng. Determine the number of unknown targets in open world based on elbow method. *IEEE Transactions on Fuzzy Systems*, 29(5):986–995, 2020.

[48] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pages 747–748. IEEE, 2020.

[49] Godwin Ogbuabor and FN Ugwoke. Clustering algorithm for a healthcare dataset using silhouette score value. *Int. J. Comput. Sci. Inf. Technol*, 10(2):32, 2018.

[50] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[51] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[52] SA Sajidha, Siddha Prabhu Chodnekar, and Kalyani Desikan. Initial seed selection for k-modes clustering–a distance and density based approach. *Journal of King Saud University-Computer and Information Sciences*, 33(6):693–701, 2021.

[53] Fuyuan Cao, Jiye Liang, and Liang Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36(7):10223–10228, 2009.

[54] Peter Roßbach. Neural networks vs. random forests–does it always have to be deep learning? *Germany: Frankfurt School of Finance and Management*, 2018.

[55] Iftikhar Ahmad, Manabu Kano, Shinji Hasebe, Hiroshi Kitada, and Noboru Murata. Gray-box modeling for prediction and control of molten steel temperature in tundish. *Journal of Process Control*, 24(4):375–382, 2014.

[56] Tae-Hwy Lee, Aman Ullah, and Ran Wang. Bootstrap aggregating and random forest. *Macroeconomic forecasting in the era of big data: Theory and practice*, pages 389–393, 2020.

[57] Philipp Probst and Anne-Laure Boulesteix. To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, 18(181):1–18, 2018.

[58] Roxane Duroux and Erwan Scornet. Impact of subsampling and pruning on random forests. *arXiv preprint arXiv:1603.04261*, 2016.

[59] Divya Pramasani Mohandoss, Yong Shi, and Kun Suo. Outlier prediction using random forest classifier. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0027–0033. IEEE, 2021.

[60] Leo Breiman. Random forests. *Machine learning*, 45:23–24, 2001.

Appendix

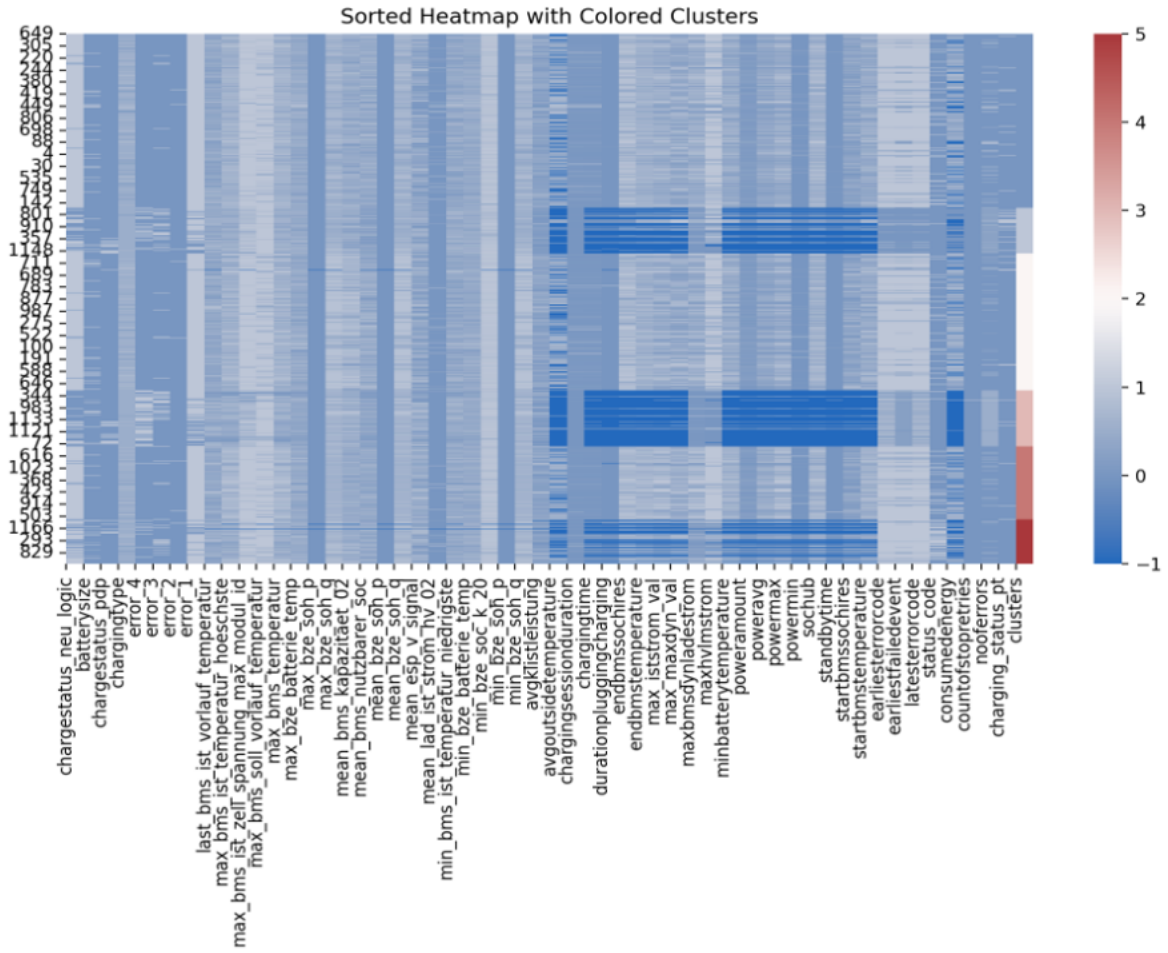


Figure 7.1: 2.png

Final heatmap image using K-Prototypes clustering algorithm is added 7.2 is rotated for an ease of reading purpose.

Error code in the Heatmap	Error name used in the Thesis
Error 1	error_detected_charging_continued
Error 2	error_detected_charging_stopped
Error 3	hvlm_fehlerstatus_error (Not used in this thesis)
Error 4	hvlm_zustand_led_error
Error 5	hvlm_hvlb_sollmodus_error (Not used in this thesis)

Table 7.1: Description of errors in the heatmap 7.2

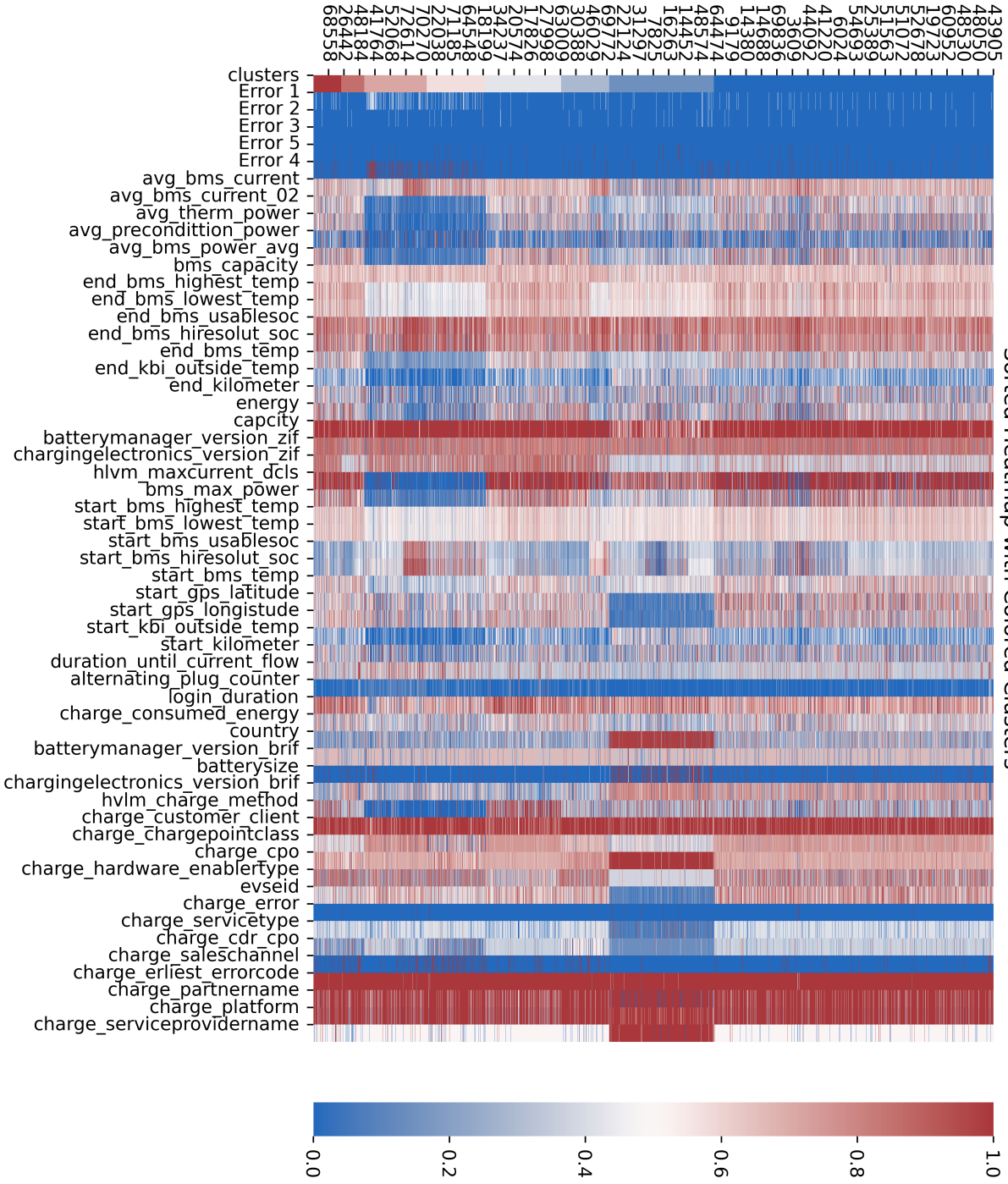


Figure 7.2: The Final Heatmap of the dataset using K-Prototype clustering algorithm

Signals or KPIs used in this Thesis
avg_bms_current
avg_bms_current_02
avg_therm_power
avg_precondittion_power
avg_bms_power_avg
bms_capacity
end_bms_highest_temp
end_bms_lowest_temp
end_bms_usablesoc
end_bms_hiresolut_soc
end_bms_temp
end_kbi_outside_temp
end_kilometer
energy
capcity
batterymanager_version_zif
chargingelectronics_version_zif
hlvm_maxcurrent_dcls
bms_max_power
start_bms_highest_temp
start_bms_lowest_temp
start_bms_usablesoc
start_bms_hiresolut_soc
start_bms_temp
start_gps_latitude
start_gps_longistude
start_kbi_outside_temp
start_kilometer
duration_until_current_flow
alternating_plug_counter
login_duration
charge_consumed_energy
country
batterymanager_version_brif
batterysize
chargingelectronics_version_brif
hvlm_charge_method
charge_customer_client
charge_chargepointclass
charge_cpo
charge_hardware_enablertype
evseid
charge_error
charge_servicetype
charge_cdr_cpo
charge_saleschannel
charge_erliest_errorcode
charge_partnername
charge_platform
charge_serviceprovidername

Table 7.2: Signals or KPIs used in this thesis

Group	Important features
Threshold Distance: 0.80	
1	duration_until_current_flow, alternamting_plug_counter, avg_precondition_power,end_kbi_outside_temp, start_kbi_outside_temp
2	charge_consumed_energy, energy, start_bms_usable_soc, start_bms_hiresolutsoc
3	start_bms_highest_temp, start_bms_lowest_temp, start_bms_temp
Threshold Distance: 1.01	
1	start_gps_lattiude, start_gps_longitude, country, charge_cpo, charge_service_providername, charge_chargepointclass, evseid, charge_cdr_cpo
2	avg_tme_verbrauch_gesamt
3	fap_battery_manager_brif, bms_capacity, end_kilometer, start_kilometer

Table 7.3: Important Features group for Error detected but charging continued

Group	Important features
Threshold Distance: 1.01	
1	start_bms_highest_temp, start_bms_lowest_temp, start_bms_temp
2	start_gps_latitude, start_gps_longitude, country, charge_cpo, charge_serviceprovidername, charge_chargepointclass, evseid, charge_cdr_cpo
3	avg_precondition_power, start_kbi_outside_temp, end_kbi_outside_temp
Threshold Distance: 1.50	
1	start_gps_latitude, start_gps_longitude, country, charge_cpo, charge_serviceprovidername, charge_chargepointclass, evseid, charge_cdr_cpo
2	charge_saleschannel
3	avg_precondition_power, start_kbi_outside_temp, end_kbi_outside_temp, charging_electronics_version_zif, charging_electronics_version_brif, charge_hardware_enablertype, duration_until_current_flow, login_duration

Table 7.4: Important Features group for Error detected and charging stopped

Group	Important features
Threshold Distance: 0.80	
1	energy, start_bms_usablesoc, start_bms_hiresolut_soc, charge_consumed_energy
2	hvlm_charge_method, hvlm_max_current_dcls, avg_therm_power, end_bms_temp, end_bms_highest_temp, end_bms_lowest_temp, bms_max_power, avg_bms_current_02, avg_bms_power_avg
3	start_gps_latitude, start_gps_longitude, country, charge_cpo, charge_service_providername, charge_chargepointclass, evseid, charge_cdr_cpo
Threshold Distance: 1.08	
1	energy, start_bms_usablesoc, start_bms_hiresolut_soc, charge_consumed_energy
2	hvlm_charge_method, start_bms_lowest_temp, start_bms_highest_temp, start_bms_temp, hvlm_max_current_dcls, avg_therm_power, end_bms_temp, end_bms_highest_temp, end_bms_lowest_temp, bms_max_power, avg_bms_current_02
3	start_gps_latitude, start_gps_longitude, country, charge_cpo, charge_service_providername, charge_chargepointclass, evseid, charge_cdr_cpo

Table 7.5: Important Features group for LED notification error

tocchapterAppendix