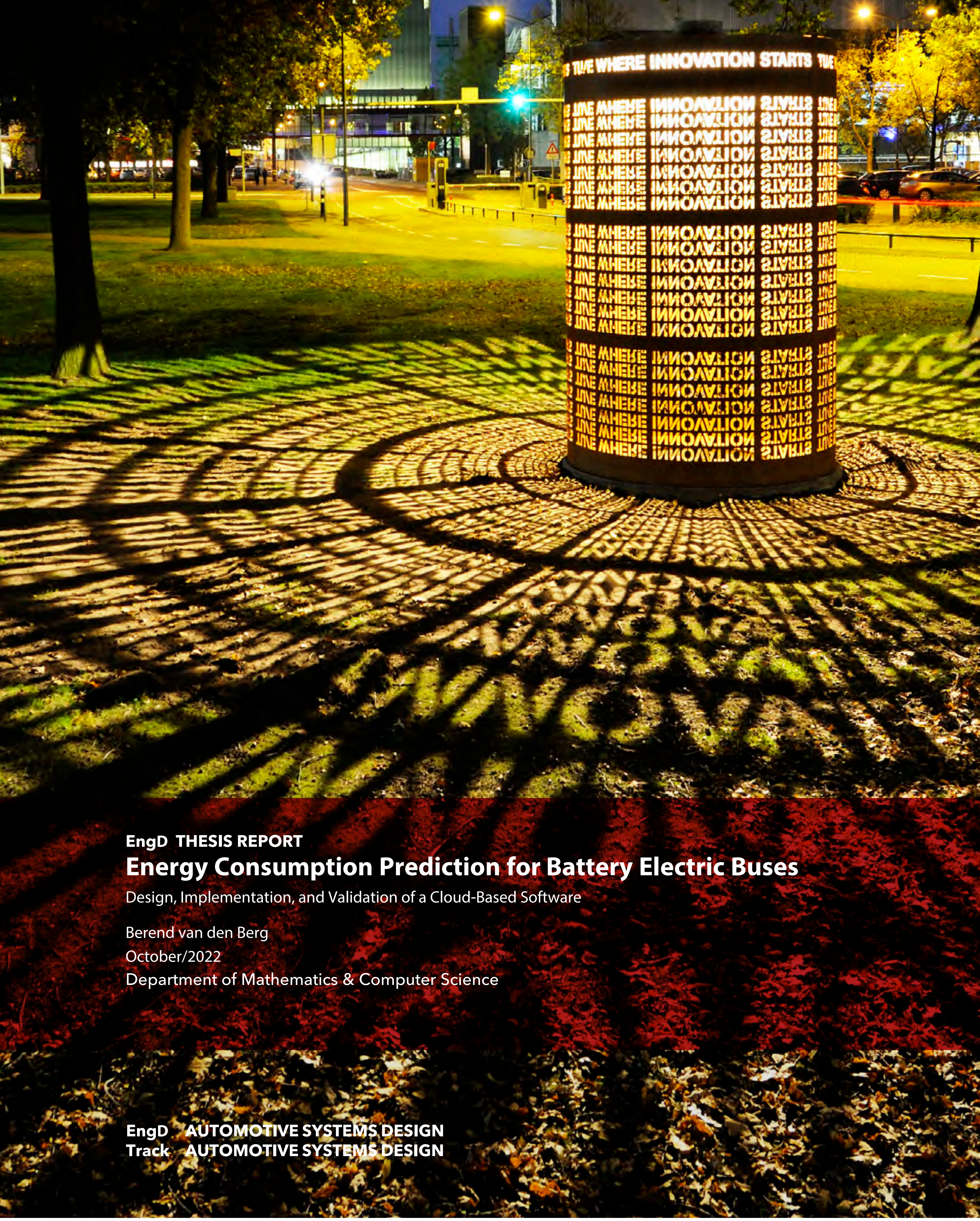


EngD THESIS REPORT

Energy Consumption Prediction for Battery Electric Buses

Design, Implementation, and Validation of a Cloud-Based Software

Berend van den Berg

October/2022

Department of Mathematics & Computer Science

EngD AUTOMOTIVE SYSTEMS DESIGN
Track AUTOMOTIVE SYSTEMS DESIGN

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Energy Consumption Prediction for Battery Electric Buses
Design, Implementation, and Validation of a Cloud-Based Software

Berend van den Berg

October 2022

Eindhoven University of Technology
Stan Ackermans Institute - Automotive/Mechatronic Systems Design

EngD Report: 2022/092

Partners

Sycada Nederland B.V.	Eindhoven University of Technology

Steering Group	dr.ir. I.J.M. (Igo) Besselink	Scientific supervisor TU/e
	dr.ir. I. (Ion) Barosan	Scientific supervisor TU/e
	R. (Rogier) Mulder BSc	Company representative Sycada
	K. (Kristian) Winge MBA	Company representative Sycada
	ir. R. (Riske) Meijer	Manager EngD ASD TU/e
Date	October 2022	

Composition of the Thesis Evaluation Committee:

Chair:	dr.ir. I.J.M. (Igo) Besselink	i.j.m.besselink@tue.nl
Members:	dr.ir. I. (Ion) Barosan	i.barosan@tue.nl
	R. (Rogier) Mulder BSc	mulder@sycada.com
	K. (Kristian) Winge MBA	winge@sycada.com
	ir. R. (Riske) Meijer	r.meijer1@tue.nl
	prof.dr.ir. D.M.J. (David) Smeulders	d.m.j.smeulders@tue.nl
	dr.ir. M.R.U. (Mauro) Salazar	m.r.u.salazar@tue.nl

Date	October, 2022
Contact address	Eindhoven University of Technology Department of Mathematics and Computer Science Automotive Systems Design MF 5.075 P.O. Box 513 NL-5600 MB Eindhoven, The Netherlands +31 402743908
Published by	Eindhoven University of Technology
EngD Report	2022/092

Abstract

The global trend of realizing zero-emission transportation involves transitioning from conventional fossil fuels to electric variants. This transition exposes public transport operators to increased risk due to reduced range and unpredictable energy consumption of their battery electric buses. This thesis presents the design of an energy consumption forecasting method, aiming to improve predictability.
A literature study was performed on state-of-the-art methods for modeling energy consumption, and the Route Characteristics Indicator (RCI) and Machine Learning (ML) methods were implemented and evaluated. Both methods' performance was compared after simulating experimental data of battery electric buses.
The presented final design splits the energy consumption estimate into two parts. The drivetrain was modeled with the RCI, and the auxiliary systems with a historical averaged auxiliary power, combined with a remaining time estimate. The design was implemented as a cloud service and verified experimentally.
The results demonstrated that the implemented software could predict energy consumption with an average error of -3%, although there was a significant spread in the estimation error, which should be reduced in future work.

Keywords	Real-time, Trip, Energy, Consumption, Prediction, Forecasting, Modeling, Battery, Electric, Urban, City, Bus, Physical, Drive Cycle, Route Characteristics Indicator, Machine Learning, Artificial Intelligence, Deep Learning (DL).
Preferred reference	Energy Consumption Prediction for Battery Electric Buses, Design, Implementation, and Validation of a Cloud-Based Software. Eindhoven University of Technology, EngD Report 2022/092, October 2022.
Partnership	This project was supported by the Eindhoven University of Technology and Sycada Nederland B.V.

Foreword

Three years ago, a group of technology pioneers in their fields of expertise set out to create an online platform to facilitate operational excellence in zero-emission public transport. This initiative was called Cloud Your Bus (CYB). The promise of CYB was to reduce capital expenses by at least 10%, reduce operational disruptions and associated costs radically, and improve the quality of service at the same time. This system is available and in operation – and it was timely because electric transport is destined for rapid growth in the years to come.

With Destination Zero, Sycada and Eindhoven University of Technology (TU/e) have taken the game even further in a project co-sponsored by the Dutch Ministry of Infrastructure and Waterways: the development of a real-time energy prediction model that can accurately forecast energy usage and battery state-of-charge at the end of a route, just shortly after a bus has commenced its journey. At Sycada, we are thrilled, and proud, that this project has led to tangible, demonstrated results.

The practical usability of the EVEE model is twofold: Firstly, it allows for early flagging of critical SoC deviations that could ruin the current day-to-day planning and hence require adaptations in the operational planning. Secondly, it allows for continuous improvements in tactical line- and charge planning by creating transparency in the impact of load, driving style, seasonality, and other environmental factors on energy usage. This creates much more forward visibility and control over operations than is available now.

EVEE hence provides another very important piece in the puzzle for the transformation towards zero-emission transportation in Europe.

Thank you, TU/e, for your guidance and support, and indeed to Berend, for a successful and rewarding cooperation.

Kind regards,

Kristian K. Winge
CEO, Sycada

October 3, 2022

Preface

This thesis is the final deliverable for obtaining the Engineering Doctorate in Automotive Systems Design at the Eindhoven University of Technology.

The presented work, titled *Energy Consumption Prediction for Battery Electric Buses*, has been conducted in collaboration with industry partner Sycada. This project is part of the DKTI-transport project Destination Zero, funded by the Netherlands Enterprise Agency, and has been a continuation of the previously conducted Cloud Your Bus project [1].

The document is written for readers with an academic or industrial background, who have technical knowledge, and are interested in energy consumption modeling for battery electric buses.

Readers who are solely interested in the project results can suffice by reading the introduction in Chapter 1, the problem analysis in Chapter 2, the final implementation and results in Chapter 6, and the conclusions and recommendations in Chapter 8.

Those interested in the theoretical background and rationale behind the final design may benefit from reading the literature survey on energy consumption modeling in Chapter 3, and the Route Characteristics Indicator and Machine Learning modeling method evaluations in Chapter 4 and 5, respectively.

Berend van den Berg

September 8, 2022

Acknowledgements

The last year was a challenging but rewarding journey, which would have been difficult to successfully complete without the much-appreciated guidance and support of those involved in the project. Therefore, I would like to use this opportunity to thank those individuals personally.

I would like to express my sincerest gratitude to the company representatives. Firstly, to Rogier Mulder, who has unceasingly supported me technically throughout the project, and to Kristian Winge, who helped place the project in a business perspective. Thank you both for your support and engaging and educational discussions.

Moreover, I am very grateful to my scientific supervisors. To Igo Besselink for his continuous support in the energy consumption modeling domain and Ion Barosan for his guidance on systems design. Special thanks to you both for the supervision and motivational discussions throughout the project, which often provided new angles for approaching my faced technical issues.

Next, I would like to thank Riske Meijer for realizing the project, Peter Heuberger for admitting me to the program, Ellen van Hoof for organizational assistance, and Yuzhe Ma for his contributions to energy consumption modeling.

At last, I would like to thank my family and close friends for motivating me to pursue and complete my Engineering Doctorate.

Thank you all for your contributions.

Berend van den Berg

September 8, 2022

Executive Summary

The global trend of realizing zero-emission transportation involves transitioning from conventional fossil fuels to electric energy. This transition exposes bus operators to increased risk in their operation due to a reduced range and unpredictable energy consumption of their fleet of battery electric buses. The previously conducted Cloud Your Bus (CYB) project aimed to reduce this risk by providing insight into the real-time operation of the fleet, realized with an embedded device sending bus data to an online platform. The operator accesses this platform.

The project Destination Zero continued this work, aiming to extend CYB with an Electric Vehicle Energy Estimation (EVEE) system capable of forecasting trip energy consumption. This estimation should improve predictability and reduce operational risks. This risk reduction may reduce the required investments by allowing the purchase of a smaller bus fleet through optimal allocation. Moreover, reducing the frequency of operational disruptions can reduce operational expenses. Both facilitate the adaption of zero-emission transportation.

To realize these benefits, an accurate and reliable energy consumption estimate should be established that is being updated while driving. Therefore, two methods for modeling the trip energy consumption have been evaluated by performing simulations with historical trip data. After a benchmark of the Route Characteristics Indicator and Machine Learning energy consumption modeling methods, the EVEE software is designed to autonomously estimate the energy and facilitate scaling to large fleets of buses.

The final design of EVEE employs the Route Characteristics Indicator modeling method, deploys as a web service, and receives data from the device mounted on the bus. In contrast to the relatively inaccurate estimate provided by the Original Equipment Manufacturer, which is based on historical energy consumption, the proposed estimation uses the repetitive and predictive nature of bus trips. The design has been implemented and validated, after collecting experimental data with 42 buses driving in a city in the Netherlands. The main results are as follows:

- The estimation error starts relatively high and reduces as the trip progresses
- The mean estimation error at 20% trip progress equals -3% for 3680 trips
- The error after 20% trip completion stays within $[-24.9, +18.7]\%$ for 95% of 3680 trips
- Estimation is done 94% of the time while driving on a schedule
- The total data transfer costs are €1.03 per bus per month.

To conclude, the validation demonstrates that EVEE is close to meeting all the requirements, and can improve the predictability of energy consumption of a battery electric bus. Consequently, it reduces the risk for the operator, and the recommendation is to continue developing the software. Several activities for further improving the estimation accuracy are proposed.

Glossary

auxiliary systems	All systems in the bus that are not used for moving the vehicle, like heating, air conditioning, hydraulics, lighting, etc. 13
AWS	Amazon Web Services 9, 39
BISON	Beheer Informatie Standaarden OV Nederland 16, 19, 24, 29
CRISP-DM	CRoss Industry Standard Process for Data Mining 23, 69
CYB	Cloud Your Bus 2, 6
drivetrain systems	Systems that consume energy for moving the vehicle, mostly the electric motor. 13
eBus	Battery Electric Bus 1, 3, 5, 6, 16
EVEE	Electric Vehicle Energy Estimation xiii, xvii, 3, 4, 6–10, 16, 25, 29, 31–36, 38–42, 45
GHG	Green House Gas 1
GoLang	Go Programming Language 4, 8, 39, 40
GPS	Global Positioning System 2, 19, 67
ICE	Internal Combustion Engine 1
IoT	Internet of Things 4
MAPE	Mean Absolute Percentage Error 25, 29, 79
ML	Machine Learning xiii, xix, 4, 14, 22, 25, 39, 41, 69, 71, 73–75, 79
MPE	Mean Percentage Error 8, 11, 20, 21, 29, 31, 39, 40, 63
NeTEx	Network Timetable Exchange 9, 11, 16, 34
OEM	Original Equipment Manufacturer 3, 9
OSM	Open Street Map 16
OWM	Open Weather Map 16

PE	Percentage Error 20
R2	Coefficient of Determination 25
RCI	Route Characteristic Indicator xiii, xv, xviii, xix, 4, 15–23, 29, 33, 38, 39, 41, 42, 63, 67
RMSE	Root Mean Square Error 7, 8, 11, 25, 40
SD	Standard Deviation 20, 31
SoC	State Of Charge 26, 75
Sycada	Sycada Nederland B.V. xiii, 40, 57
SysML	Systems Modeling Language 31, 33–38
TU/e	Eindhoven University of Technology xiii, 57
ZE	Zero-Emission 1, 2, 9, 33

List of symbols

	Energy consumption modeling
a	Acceleration
A	Frontal area
C_r	Rolling resistance coefficient
C_d	Aerodynamic drag coefficient
D_r	Remaining distance
E	Energy
F	Force
g	Gravitational constant
G_a	Auxiliary power correction gain
m	Mass
m'	Estimated mass
N	Normalized energy consumption prediction
P	Power
RCI	Route Characteristics Indicator
s	Distance
t	Time
v	Velocity (longitudinal)
w	Wind speed (in driving direction)
W	Weight factor
η	Efficiency
ψ	Road inclination angle
ρ	Air density
	Kalman filter
y	Observed measurement
$\hat{y}$	Estimated measurement
Ψ	Gradient of model output
θ	Parameter to estimate
$\hat{\theta}$	Estimated parameter
e	Error
P	Error covariance matrix
R	Constant covariance matrix
K	Kalman gain

List of Tables

List of Figures

Contents

1 Introduction

1.1 Zero-Emission transportation

Global warming has become an important issue, addressed by governments all over the world by instating laws aiming to reduce the amount of expelled Green House Gas (GHG). Figure 1.1 shows the projected energy consumption and GHG emission estimated by the European Commission, where the transport sector is a considerable contributor. Therefore, the aim is to realize Zero-Emission (ZE) transportation by replacing fossil fuels with ZE variants. These measures stimulate the adaptation of alternative drivelines.

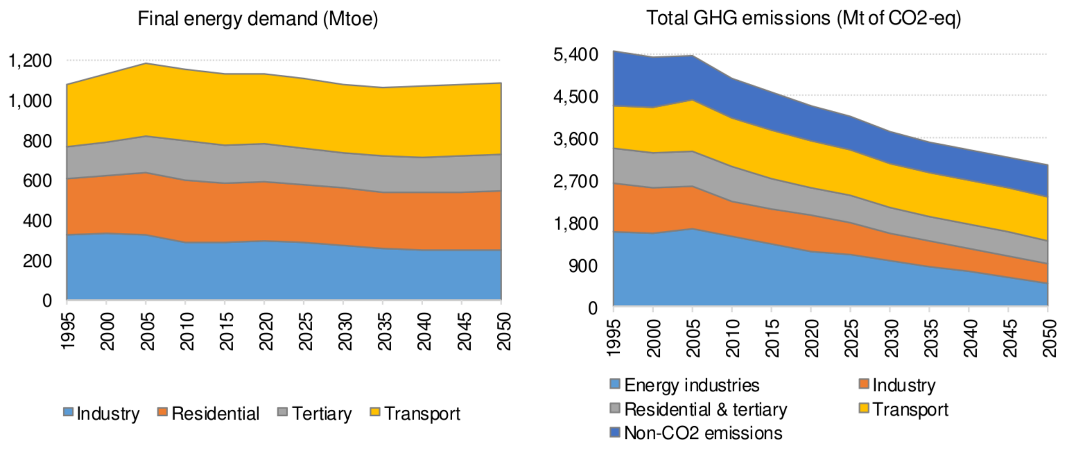


Figure 1.1: Evolution of total energy consumption and GHG emissions between 1995 and 2050 [2].

A historic overview of the volumes of alternative drivelines for heavy-weight vehicles is presented in Figure 1.2. Here, significant growth can be observed in the adaptation of the Battery Electric Bus (eBus). This introduces many challenges, which among others include:

1. **Reduced range:** in contrast to fossil fuel-powered buses - which have enough range to drive a full day without refueling - the eBus has significantly less range. Recharging during the day introduces complex scheduling tasks.
2. **Range uncertainty:** with the introduction of batteries, a range uncertainty is introduced due to battery degradation, resulting in a decreasing capacity and thus range over time.
3. **Less predictable energy consumption:** For example, using electric heating instead of heat from the Internal Combustion Engine (ICE) introduces different energy usage patterns.

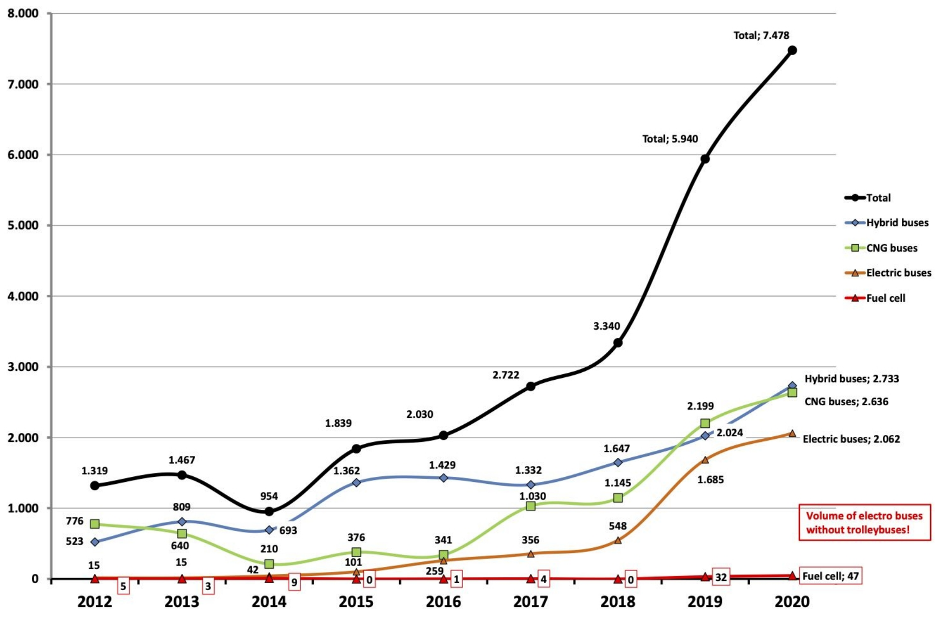


Figure 1.2: Volumes of alternative drivelines in Western Europe and Poland with a gross vehicle weight over 8 tons [3].

1.2 The Cloud Your Bus project

The previously conducted Cloud Your Bus (CYB) project [1] addresses these challenges by offering a vehicle telematics system and a range of Zero Emission (ZE) services. These services aid the public transport operators in deploying their fleet of electric buses optimally, by reducing the risks and costs of the operation. The following work packages were defined:

1. **Data taxonomy:** streaming live data to the operator
2. **Live charging data:** integrating charging points in the operational planning
3. **Driving style optimization:** reducing energy consumption through real-time driver feedback on an in-vehicle display
4. **Planning optimization:** optimizing fleet allocation with dynamic re-planning during disruptions
5. **Battery state monitoring:** monitoring the battery state-of-health
6. **Energy consumption forecast:** predicting trip energy consumption during operation to optimize schedules and detect charge deficiencies early during operation

An overview of the realized CYB system is presented in Figure 1.3. Here, an embedded device communicates with the bus through a CAN-bus connection and collects vehicle and Global Positioning System (GPS) data. The device sends the data to several cloud services, to which the operator has access, see Figure 2.3. Two generations of devices are currently in use by operators. Approximately 90% are Owasys OWA3x devices and 10% the new generation Owasys OWA450. New buses will be equipped with the new generation device, which has extended computational, storage, and connectivity capabilities compared to the old device.

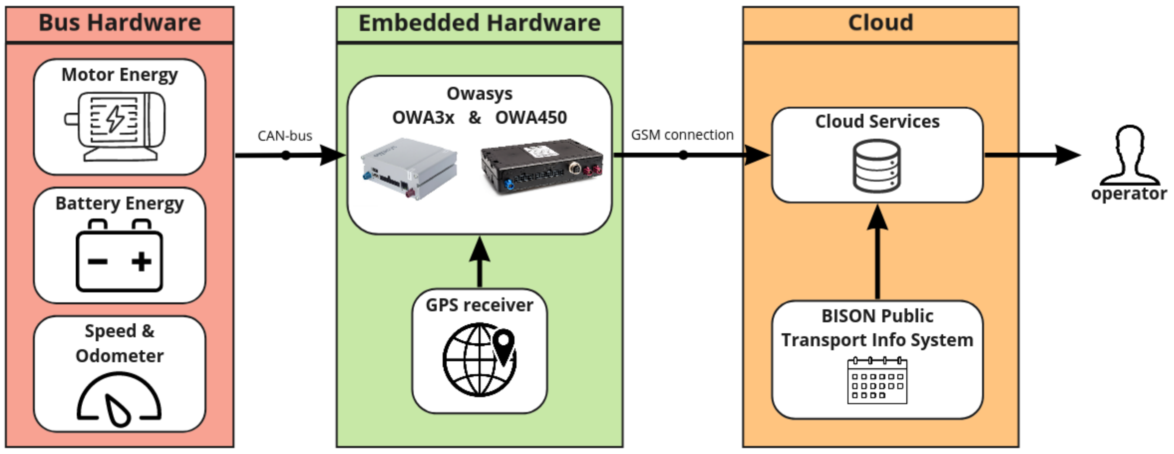


Figure 1.3: Cloud Your Bus system realization. The device collects GPS and vehicle data and sends this to the cloud services, to which the operator has access.

1.3 Destination Zero: Electric Vehicle Energy Estimation

The work in this thesis is part of the DKTI-transport project Destination Zero, funded by the Netherlands Enterprise Agency, and presents an Electric Vehicle Energy Estimation (EVEE) system. This addresses work package 6: energy consumption forecast, with the main goal to:

Develop, validate, and productize an Electric Vehicle Energy Estimation system, that can provide a forecast of the total trip energy consumption for battery electric city buses, at any point throughout the trip as accurately as possible.

Currently, an estimate may be provided on the driver dashboard by the Original Equipment Manufacturer (OEM), which is based on average historical energy consumption, but this is insufficiently inaccurate for predictions. Realizing EVEE would reduce the uncertainty in the energy consumption of the eBus. Therefore, the operator can reduce their vulnerability, and improve the deployment of their fleet through optimized planning. Hence, the **scope** of this project is defined as follows:

1. Analyzing the energy consumption forecasting problem
2. Modeling the energy consumption:
   - Validating the previously developed prototype energy consumption model [1]
   - Developing and validating a data-based Machine Learning energy consumption model
   - Selecting the best modeling approach through a benchmark
3. Designing and implementing the EVEE software
4. Verifying and validating the implementation through real-world experiments

Assumptions and Constraints

- EVEE shall estimate energy consumption for battery electric city buses operating in public transport, which drive along fixed schedules and routes
- EVEE shall work with the Owasys OWA3x and OWA450 Internet of Things (IoT) devices
- Go Programming Language (GoLang) shall be primarily used during implementation

1.4 System development methodology

The system development methodology employed during the development of EVEE, is presented in Figure 1.4. For a more detailed description of the process, see the project management plan attached in Appendix A.

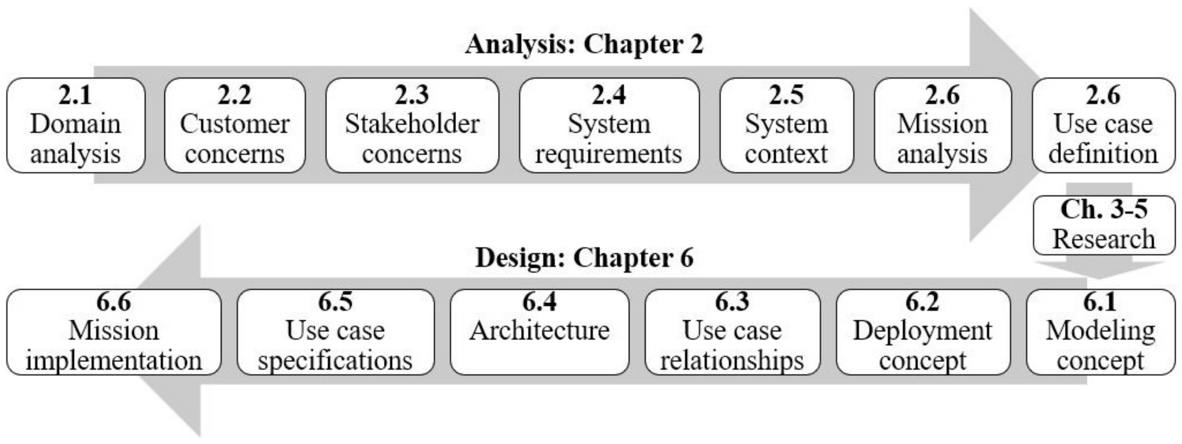


Figure 1.4: Employed system development methodology

1.5 Report outline

The report is organized as follows. First, in Chapter 2 the analysis of the problem and domain are discussed. Next, Chapter 3 presents an overview of state-of-the-art methods for energy consumption modeling, and two methods are selected for further analysis. These are evaluated in Chapters 4 and 5, the Route Characteristics Indicator (RCI) and Machine Learning (ML) methods respectively. Chapter 6 describes the design phase, including the selection of a modeling method and a software deployment concept, followed by a presentation of the software design. Chapter 7 discusses the implementation and validation. Finally, Chapter 8 provides the conclusions and recommendations.

2 Analysis of the problem

This chapter describes the analysis phase of the development methodology (see Figure 2.1), followed by an exploration and evaluation of energy modeling concepts in Chapters 3-5, and the design phase in Chapter 6.

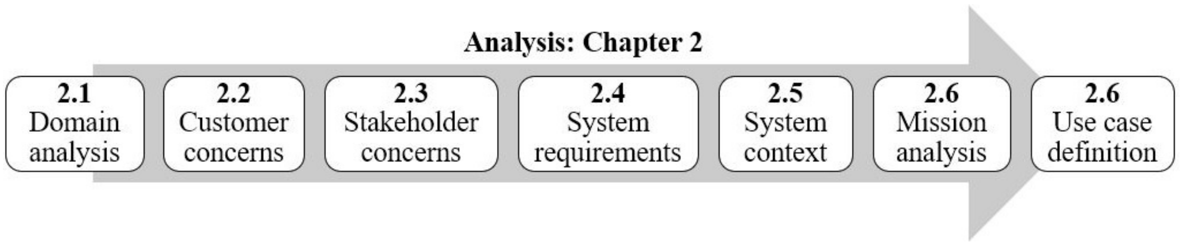


Figure 2.1: Development methodology: analysis.

2.1 Domain analysis

The domain analysis in this section is vital for designing an effective solution to a problem, since the nature of the solution may vary between domains. The considered domain is buses operating in public transport, which drive with fixed schedules and routes (often called lines). Consequently, the same sequence of trips is driven repeatedly throughout the year. This repetitive nature can aid the forecasting of the energy consumption for these schedules.

The schedule of an eBus consists of several blocks of trips, which start at the depot, followed by several trips, and ends back at the depot for charging. Figure 2.2 illustrates an example where the bus leaves the depot, drives to the central bus station, visits destinations A and B, and returns to the depot.

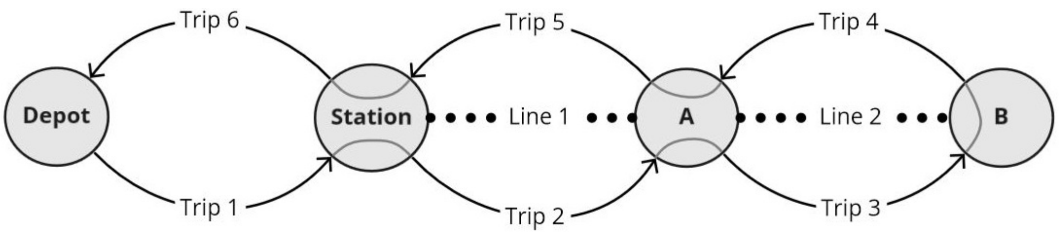


Figure 2.2: Operating schedule context example. This figure illustrates one of many blocks of a schedule. Lines are physical routes that connect destinations. Trips are unique instances that the bus drives along these lines.

EVEE shall initially operate in the Netherlands, where schedules are updated approximately twice a year. The operator constructs these schedules based on many variables, including the estimated range of the eBus.

2.2 Customer concerns

The customer concerns presented in this section are leading during the design generation. After an extensive problem analysis in consultation with stakeholders, the identified concerns were reducing the operation's costs, vulnerability, and carbon footprint. The CYB system addresses these concerns through four different value streams, presented in Figure 2.3. EVEE shall extend this system with real-time energy consumption estimation, providing the following additional customer value:

1. Operators implement battery capacity buffers in their schedules to cope with the energy consumption uncertainty, resulting in buses returning to the charger with up to 25% charge remaining. Reducing the uncertainty allows for smaller buffers, leading to better use of material, and allowing filling the same schedule with fewer buses. Thus, reduced buffers require fewer investments and operational costs.
2. Operators often resolve operational disturbances reactively due to a lack of predictive methods. A reliable energy estimate can be a method to implement predictive problem resolution. For example, when a bus cannot return to the charger due to excessive energy usage and not enough remaining charge. The reactive approach may detect and react once the problem has become unresolvable, leading to a stranded bus. However, the predictive method could identify this timely, allowing swapping of the bus and avoiding stranding.
3. The monitoring and schedule optimization platforms can be extended with the energy forecast, allowing the operator to react to disturbances predictively in two ways. This may be done by monitoring the fleet, and through the adaptive schedule optimization software.
4. The driver feedback algorithm may be improved by including deviations from the planned energy consumption.

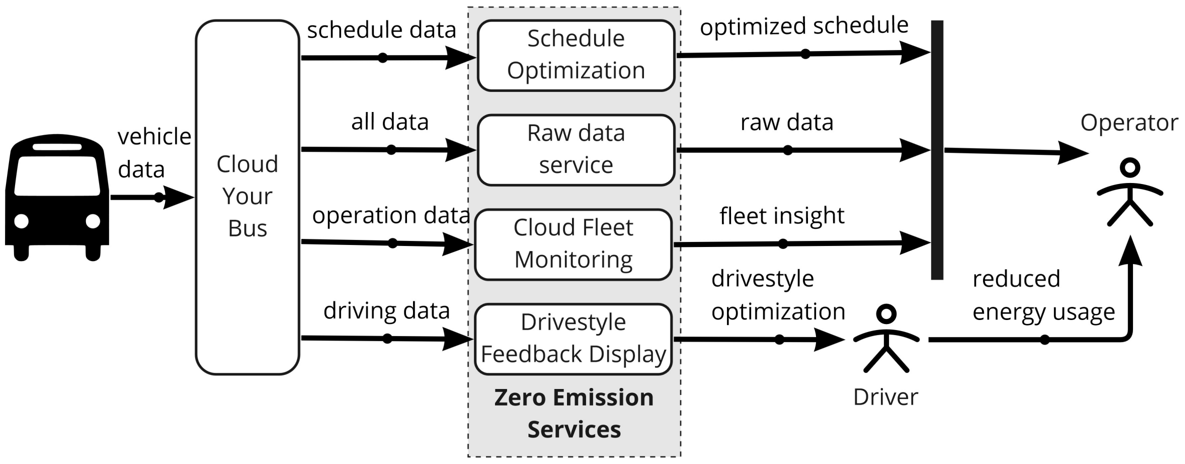


Figure 2.3: Value streams of the Cloud Your Bus system.

2.3 Stakeholder concerns

A stakeholder analysis was performed to identify communication preferences and their concerns regarding the project, which is described detailed in Appendix B. Their main concerns regarding the system are summarized in Table 2.1.

Table 2.1: Stakeholder concerns regarding the energy estimation.

ID	Keyword	Concern
C1	Estimation	Provide a reliable energy estimate with EVEE
C2	Costs	Minimize total costs of operating EVEE
C3	Ease-of-use	Minimize effort to use and operate EVEE
C4	Maintainability	Monitor and maintain EVEE easily
C5	Compliance	Compliance of EVEE with company and government policies
C6	Scalability	Ability to scale EVEE by deploying to many devices

2.4 System requirements

The system requirements were elicitated from the stakeholder concerns and grouped into functional and non-functional requirements, see Table 2.2 and Table 2.3 respectively. The non-functional requirements include the quality aspects of the functional requirements. Both are prioritized using the MoSCoW[1] method, which includes the following classes: must have, should have, could have, and won't have. To ensure the generated requirements address the concerns, a requirement tracing chart was established, which can be found in Appendix C.

Table 2.2: Functional requirements of EVEE, defining the high-level system functionalities, and prioritized with the MoSCoW method.

ID	Keyword	Requirement	Priority	Addresses
FR-1	Estimation	The EVEE system must provide an estimate of the total energy consumption in kWh for a trip.	Must	C1
FR-2	Ease-of-use	The EVEE system must boot-up and configure itself for use with any battery electric city bus in Sycada's portfolio automatically on bus power-up.	Must	C3
FR-3	Monitoring	The EVEE system should provide the Root Mean Squared Error (RMSE) of the trip energy consumption estimate, for the maintainer to monitor the operating performance.	Should	C4

[1]Wikipedia, MoSCoW method, accessed 28/7/2022 at https://en.wikipedia.org/wiki/MoSCoW_method

Table 2.3: Nonfunctional requirements of EVEE, prioritized with the MoSCoW method. Including: Scalability, Reliability, Regulatory, Maintainability, Serviceability, Utility, Security, Manageability, Data integrity, Capacity, Availability, Usability, Interoperability, and Environmental requirements.

ID	Keyword	Requirement	Priority	Addresses
NF-1-A	Speed	The EVEE system must publish the energy consumption estimate every 55 to 65 seconds.	Must	FR-1
NF-1-B	Accuracy	The EVEE system should estimate the total trip energy, with less than 25% Mean Percentage Error (MPE), after 20% of the traveled trip distance, for at least 95% of at least 500 trips.	Should	FR-1
NF-1-C	Uptime	The EVEE system should provide an energy estimate at least 99% of the time while on a schedule.	Should	FR-1
NF-1-D	Initialization	The EVEE system should provide its first energy estimate within two weeks after a schedule revision.	Should	FR-1
NF-2-A	Costs	The total costs of operation of EVEE must not exceed €5 per bus per month.	Must	C2
NF-2-B	Bandwidth	The bandwidth usage of the EVEE system must be less than 0.05 kilobytes per second per bus, averaged over a minimum of 1 week.	Must	NF-2-A
NF-3-A	System start	Initialization and configuration of the EVEE system should take no longer than 30 seconds after power-up.	Must	FR-2
NF-4-A	Monitoring	The EVEE system should publish the energy consumption estimate RMSE at least once per trip.	Should	FR-3
NF-5-A	Maintainability	The EVEE system should be programmed in GoLang.	Should	C4
NF-6-A	Compliance	The EVEE system must comply with the General Data Protection Regulation of the European Union.	Must	C5
NF-7-A	Scalability	The EVEE system should be compatible with all OWA3x and OWA450 devices of Sycada.	Must	C6

2.5 Context of EVEE

The context of EVEE was established, defining the system boundary and all actors that interact with the system, see Figure 2.4. Here, the driver and operator interact indirectly with the system through the Zero Emission Services. The maintainer monitors EVEE through the Amazon Web Services (AWS) CloudWatch platform.

The embedded device collects data from the sensors provided by the OEM. This data is sent in a request to EVEE, which responds with the estimate. The distribution of the estimate to the ZE services is realized through an existing interface. The KV6 server provides the real-time trip status of the vehicle, e.g. trip/line ID, and the NeTEx server provides the schedule.

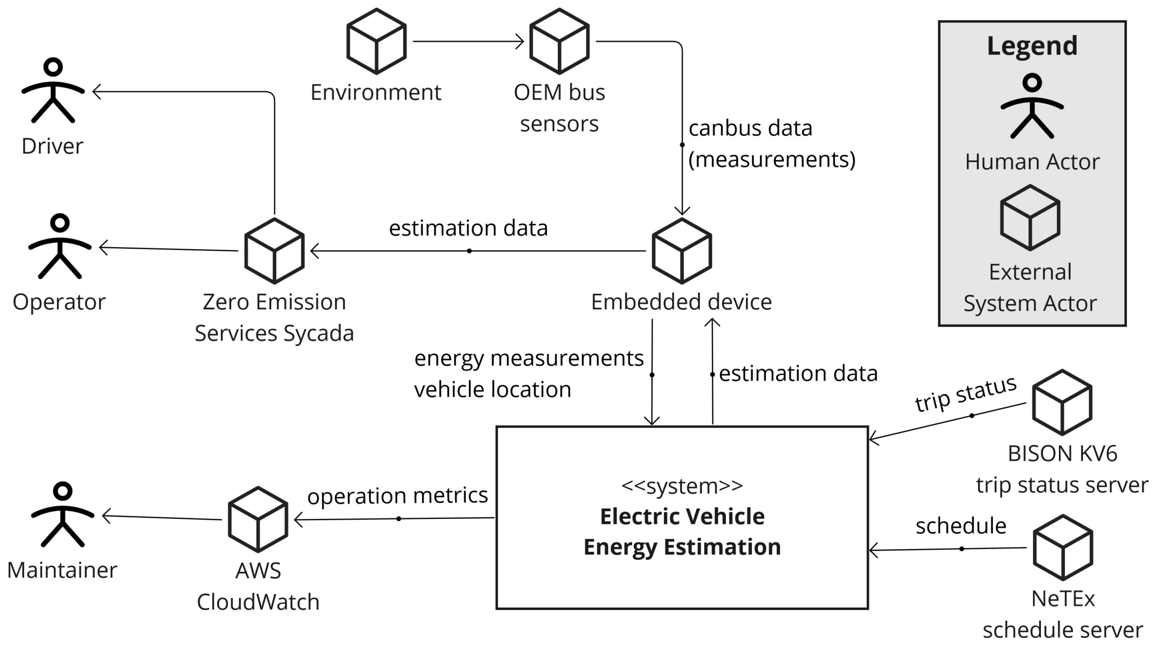


Figure 2.4: EVEE system context diagram.

2.6 Mission analysis

EVEE's main functionalities have been identified by applying a mission analysis, which aims to identify use cases for each operating phase, see Table 2.4. These use cases are the foundation for the system design introduced in Chapter 6, which extends the use cases in the table that are denoted with a + symbol, followed by the added specification.

Figure 2.5 shows the analysis for the mission: to calculate a trip energy estimate and send this to the Zero Emission cloud services. Here, the top row shows the mission phases, the middle row a timeline with events, and the bottom row the use cases per phase. The mission repeats itself by starting at a depot, driving to the start of the first public trip, driving the scheduled trips, returning back to the depot, and connecting the bus to the charger.

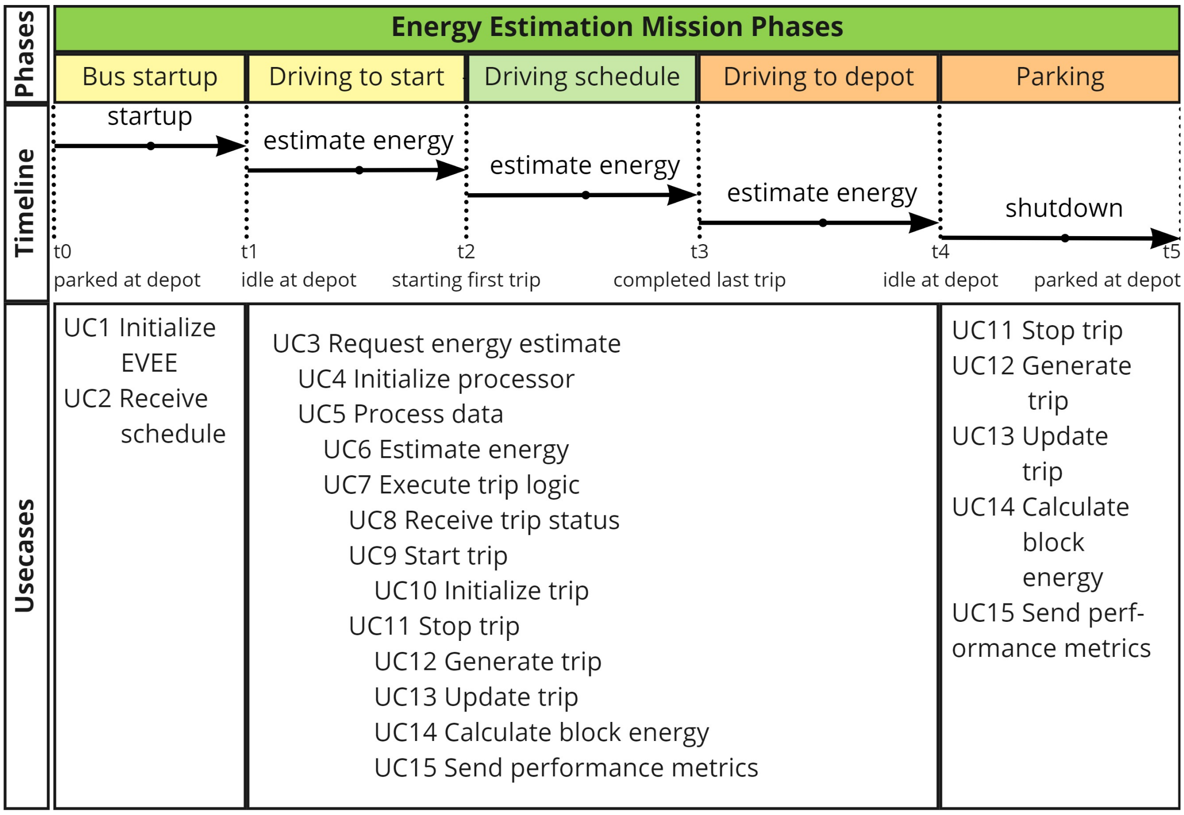


Figure 2.5: Analysis of the energy estimation mission.

2.7 Discussion

To realize EVEE's mission, an energy consumption model should be realized that satisfies the defined requirements. Therefore, the upcoming Chapters 3 to 5 explore and evaluate state-of-the-art modeling methods.

Table 2.4: Description of use cases derived from the mission analysis.

ID	Name	Actors	Description
UC1	Initialize EVEE	EVEE	Initialize EVEE when the device boots by setting up required configurations for data storage and communication. It starts receiving trip status messages and listening to estimate requests.
UC2	Receive schedule	EVEE NeTEx	EVEE receives the operator schedule by communicating with the NeTEx server.
UC3	Request energy estimate	Device EVEE	The embedded device sends a request, that EVEE responds to with a trip energy estimate. + The request contains data collected over one minute, with timestamp, latitude, longitude, speed, odometer, battery energy, and drivetrain energy.
UC4	Initialize processor	EVEE	EVEE initializes a processor for a bus.
UC5	Process data	EVEE	Execute the trip logic, update the energy measurements, append the trip data log, and update the energy estimate.
UC6	Estimate energy	EVEE	Estimate the total trip energy consumption. + This uses an RCI profile with mass estimation for the drivetrain energy, and a time remaining profile with mean historical power for auxiliary energy estimation.
UC7	Execute trip logic	EVEE	On receiving a request, changes in trip/line are identified, triggering the required start trip or stop trip use case.
UC8	Receive trip status	EVEE BISON	Receive the actual trip status from the BISON KV6 server.
UC9	Start trip	EVEE	Initialize the estimation for the current trip. + Initialize trip, configure the mass estimator with the vehicle mass, and start trip data logging.
UC10	Initialize trip	EVEE	+ Initialize a reference trip from an existing reference, or initialize one from historical trips.
UC11	Stop trip	EVEE	Processes all data collected throughout the trip. + Do the following: 1. save a datalog for debugging, 2. generate a new trip, 3. compare new trip quality with reference, 4. update reference trip, and 5. calculate the estimation error with measured energy.
UC12	Generate trip	EVEE	+ Generate a new reference trip from logged data.
UC13	Update trip	EVEE	+ Update the loaded reference trip with the newly generated one.
UC14	Calculate block energy	EVEE	Calculates the energy estimate for a series of trips that form a schedule block. It receives the schedule from the NeTEx schedule server to realize this.
UC15	Send performance metrics	EVEE Cloudwatch	On trip completion, the actual energy usage is used to calculate the estimation's MPE at every 10% progress of the trip. This is written to a log file for analysis by the maintainer. The RMSE of the estimation is sent to the AWS CloudWatch monitor.

3 State-of-the-art methods for energy consumption modeling

State-of-the-art methods for the energy consumption estimation of electric vehicles can be divided into two common approaches. Firstly, physical modeling approaches such as those in [4–8], construct a vehicle model by estimating its physical parameters, and combine this with a drive cycle to evaluate the model. Secondly, data-driven methods use historical or real-time data to train a model that best fits the data [9–16]. For an extensive review of literature on both approaches the reader is referred to [17], and for a review on data-based energy estimation methods to [13, 14]. The remainder of this chapter discusses physical models and data-based models in more detail.

3.1 Physical Modeling

The physical modeling approach works with two main components for calculating the energy consumption estimate. A drive cycle is constructed to model the trip. This is combined with a vehicle model to virtually drive the bus along this drive cycle, while calculating the energy consumption.

3.1.1 Vehicle energy consumption model

A commonly used energy consumption model in literature is composed of two terms that model drivetrain systems and auxiliary systems seperately, see (3.1) [15]. The drivetrain component models the energy required for moving the vehicle. The auxiliary component all secondary energy consumption, such as heating, air conditioning, lighting, etc. The total energy required is given by

$$E_{total} = \int_0^t (P_{drive} + P_{aux})dt, \tag{3.1}$$

where, E_{total} is the total energy consumption, P_{drive} the powertrain power, P_{aux} the auxiliary power, and t is time. P_{drive} can be further decomposed as (3.2) [15], where F_{drive} is the drive force, v is the longitudinal velocity, and η the total powertrain efficiency,

$$P_{drive} = \frac{F_{drive}v}{\eta}. \tag{3.2}$$

Next, the driving force (F_{drive}) is determined using a longitudinal dynamics model, which includes forces due to rolling resistance, road inclination, aerodynamic drag, and acceleration. These contributions are given in (3.3), where C_r is the roll resistance coefficient, m total vehicle mass including passengers, g the gravitational acceleration, ϕ the road inclination angle, ρ the air density, C_d the drag coefficient, w the wind speed in the driving direction, m_r the equivalent mass of the rotational inertia, and $a(t)$ the longitudinal acceleration.

$$F_{drive}(t) = C_r mg \cos\phi(t) + mg \sin\phi(t) + \frac{1}{2}\rho C_d A[v(t) - w(t)]^2 + [m + m_r]a(t) \tag{3.3}$$

Evaluating the presented physical model requires many parameters (m, C_d, A, $w(t)$, η), which is an exhaustive process for the diverse fleet of buses in Sycada's portfolio. Hence, applying such a model should use the real-time estimation of required parameters.

3.1.2 Drive cycle prediction

After constructing a physical model, a drive cycle should be established to calculate the energy consumption forecast for a trip. This cycle models the trip as speed and acceleration over time, and several approaches exist for determining such as cycle. Firstly, this could be created by collecting historical data on a route, which is trivial in public transport, since vehicles drive fixed routes. Alternatively, a drive cycle generator can be used [5, 7, 10, 18], or standardized drive cycles can be adopted.

3.2 Data-driven methods

Data-driven methods, such as Machine Learning (ML), avoid the physical model and instead derive a model from historically collected data. Compared to drive cycle-based methods, data-driven methods often rely on an abstraction of the driving profile. A simplified representation of the driving profile can be obtained by collecting topological parameters such as the number of stops, road condition, route length, and operational factors such as trip duration, and average speed [11, 13, 14].

The performance of data-based algorithms can be affected substantially by their input parameters. Therefore, an analysis was done to identify these parameters, and compare them with findings in the literature in Appendix D. It shows per parameter if a significant impact was noticed in the literature.

3.3 Route Characteristics Indicator method

Approaches employing a physical model require many model parameters, which may be challenging to obtain for many bus types. Moreover, data-driven methods require significant amounts of data for training a model, leading to long initialization times. Therefore, a new approach was proposed in [1], aiming to realize a generic energy consumption estimation algorithm that initializes quickly.

3.3.1 Route Characteristics Indicator

The approach aims to simplify the longitudinal dynamics equation (3.3), so it depends on parameters that can be estimated during operation. In this equation, mass is the dominant parameter and varies due to the number of passengers on the bus. This fluctuation is significant compared to the vehicle's empty weight and therefore has a big impact on the total energy consumption. Hence, the aim is to estimate the mass, by extracting it from the equation:

$$\frac{F_{drive}(t)}{m} = \underbrace{C_r g \cos\phi(t) + g \sin\phi(t) + a(t)}_{\text{route specific}} \underbrace{\frac{m + m_r}{m} + \frac{1}{2m}\rho C_d A [v(t) - w(t)]^2}_{\text{negligible}} \tag{3.4}$$

Here, the first part of equation (3.4) no longer depends on mass, and is route specific. Additionally, m is assumed to be larger than the equivalent mass of inertia of the rotational parts m_r, and therefore varies little. Moreover, aerodynamic drag is small compared to other terms since the considered buses drive in cities and therefore at low speeds. Both these terms are therefore neglibile compared to the route specific part. Thus, the right side of the equation is the characterization of a route and is called the Route Characteristics Indicator (RCI).

3.3.2 Real-time mass estimation

The RCI has the same unit as acceleration, m/s^2, and can be used to estimate the driving force using only a mass estimate. The drivetrain energy consumption estimate is calculated as:

$$RCI(s) = \frac{F_{drive}(s)}{m\eta} \approx \frac{P_{drive}(s)}{mv(s)}, \tag{3.5}$$

$$E_{drive}(s_1) = m'(s_1) \int_{s_1}^{s_2} RCI(s) ds, \tag{3.6}$$

where E_{drive} is the drivetrain energy consumption estimate from the current position s_1, to a future position s_2. The vehicle mass m is replaced with the imaginary mass m', estimated while driving using a linear Kalman filter [19], and indirectly accounts for external factors such as driving style. Furthermore, η is the powertrain efficiency, P_{drive} is the powertrain power, and v is the longitudinal velocity.

3.3.3 Total energy consumption estimate

The total energy consumption estimate calculation is presented in (3.7). It estimates the energy consumption from (t_1, s_1) to the end of the trip, (t_{end}, s_{end}). The second term on the right side is the auxiliary estimate, which includes the auxiliary correction gain G_a, which corrects the auxiliary power P_{aux} in real-time using a linear Kalman filter [19].

$$E(t_1) = m'(s_1) \int_{s_1}^{s_{end}} RCI(s) ds + \int_{t_1}^{t_{end}} G_a(t_1) P_{aux}(t) dt \tag{3.7}$$

3.4 External information sources

In literature, vehicle data is often combined with external sources. This is done to include external factors, like the environment, in the estimate. This section presents a brief review of these sources.

The environment can have a significant impact on the energy consumption estimate, thus in [5, 7, 8] Open Street Map (OSM) is employed for extending the dataset with route topology information. Moreover, road elevation and inclination can be obtained through Shuttle Radar Topography Mission (SRTM) [7], or detailed information such as traffic lights and speed bumps can be included through local systems, like Brussels UrbIS [10]. Besides topological information, also weather data can be included through platforms such as the OpenWeatherMap (OWM) [5, 7, 8],

The operational domain analysis introduces the repetitive nature of public transport trips, which can be used to the advantage of the estimation [6, 11]. It may be interesting to investigate possibilities for obtaining this data in the Netherlands since this is where EVEE shall initially operate.

Public transport operations information in the Netherlands can be accessed through Beheer Informatie Standaarden OV Nederland (BISON), which facilitates information sharing within the public transport sector. Information such as schedules, trips, lines, and stops can be obtained through the BISON KV6 [20]. On the European level, this may be obtained through the Network Timetable Exchange (NeTEx).

3.5 Discussion

The literature survey in this chapter discusses several energy estimation methods, ranging from physical modeling to purely data-based methods, each having its unique qualities. According to requirement NF-7-A, EVEE must be usable with any eBus in Sycada's portfolio, thus many different bus configurations shall be usable with the estimation method. Using a physical model requires many parameters which are not provided by the OEM, and must therefore be measured or estimated. This can be difficult to obtain. Hence, two promising approaches will be compared. The RCI and data-based methods are evaluated in Chapter 4 and 5 respectively. A comparison of both methods can be found in Chapter 6.

4 Modeling approach 1: RCI method

Prior to this project, a prototype of the RCI method was developed, which is discussed in detail in [1], and summarized in Section 3.3. The model was tested with approximately 50 trips on a single route. No experiments were done to validate this method for a large number of trips on various routes. This chapter proposes several improvements to the algorithm and presents a large-scale evaluation of the RCI method.

4.1 Updated energy estimation method

The simulation results of the method proposed in [1], attached in Appendix E, reveal that the auxiliary energy estimation error peak up to 250%, and have a significant impact on the total error. This causes the total estimation error to exceed the accuracy requirement NF-1-B, which requires less than 25% error after 20% progress. The following improvements are proposed when compared to [1]:

1. The drivetrain and auxiliary estimates, described by (4.2) and (4.3) respectively, are now composed of a measured and an estimated part. The first term is the energy measured from the trip start t_0 to time t_i, and the second term is the estimate from time t_i until the trip end.
2. The auxiliary energy estimation in (4.3) is simplified by replacing the Kalman filter with a time remaining estimate and a historically averaged auxiliary power, discussed in Section 4.1.1.

A graphical overview of the updated method is presented in Figure 4.1. The method is described by the following equations:

$$\begin{aligned} E_{total}(t_i) =& E_{drive}(t_i) && + E_{aux}(t_i), && (4.1)\\ E_{drive}(t_i) =& \int_{t_0}^{t_i} P_{drive}(t)dt + m'(s_i)\int_{s_i}^{s_{end}} && RCI(s)ds, && (4.2)\\ E_{aux}(t_i) =& \underbrace{\int_{t_0}^{t_i} P_{aux}(t)dt}_{Measured} && + \underbrace{\bar{P}_{aux}t_{rem}(s_i)}_{Estimated}, && (4.3) \end{aligned}$$

where t is the time, s is the distance, E_{total}, E_{drive}, and E_{aux} are the trip energy consumption estimates, t_i and s_i are the time and distance at which the estimate is made, t_0 is trip start time, s_{end} is the estimated trip length, RCI is the Route Characteristics Indicator, P_{aux} and P_{drive} the measured drivetrain and auxiliary power, $\bar{P}_{aux}$ is the historical average auxiliary power, and t_{rem} is the estimated time remaining.

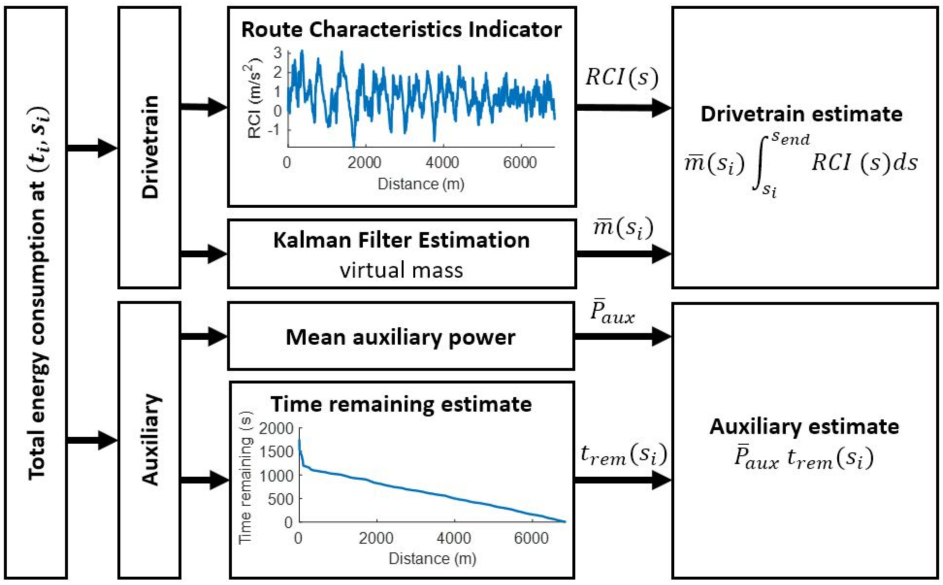


Figure 4.1: Overview of the estimation part of the updated RCI method.

4.1.1 Updated auxiliary energy estimation

Two challenges present themselves in the auxiliary estimate. Firstly, the auxiliary power highly varies throughout the day and is challenging to predict accurately. Secondly, the remaining time is unknown and varies due to traffic conditions. The proposed adjustments simplify this model by assuming a constant power throughout the trip, and modeling the remaining time as a function of traveled distance, both as historical average. This is described in Equation 4.4.

$$\int_{t_i}^{t_{end}} P_{aux} dt \approx \bar{P}_{aux} t_{rem}(s_i) \tag{4.4}$$

4.1.2 Reference trip

A reference trip is used for the estimation, which consists of several components. Firstly, it contains three arrays for the traveled distance, the RCI, and the remaining time profile. These data points are spaced equally through interpolation, which is done every 5 meters in the current implementation. Additionally, a single value for the mean auxiliary power and the mass final estimate.

The reference trip is initialized by averaging historical trips, after which the reference is updated after each new trip. Updating the reference trip allows the method to account for environmental changes such as seasonal influences on energy consumption. This is done by calculating the weighted average of the reference trip and the new trip:

$$T_{updated} = \frac{W_{update} T_{reference} + T_{new}}{W_{update} + 1}, \tag{4.5}$$

where $T_{updated}$ is the updated reference trip, $T_{reference}$ is the reference that will be updated, T_{new} is a newly generated trip, and W_{update} is the update weight. This work uses $W_{update} = 30$.

4.2 Validation approach

The results presented in [1] were established with 50 trips of a single route, of which 30 were used for initial reference generation, and 20 for driving simulations. To draw a confident conclusion about the method's accuracy, this number should be increased up to a minimum of 500 trips according to requirement NF-1-B. The goal of the validation described in this report is to extend the work done previously, by increasing the total number of trips and testing on different routes.

The model validation methodology is described in Figure 4.2. It starts with collecting a dataset consisting of vehicle data and schedule information, followed by the extraction of data segments, grouping these per route, and sorting them in a training and validation data set. The training data is used for initializing the reference trip and the validation data for simulations. The final step is to process the results into graphs and evaluate the performance. Each of the steps in the process will be discussed in the following sections.

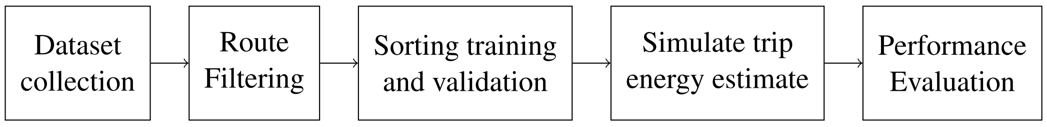

Dataset collection
Route Filtering
Sorting training and validation
Simulate trip energy estimate
Performance Evaluation

Figure 4.2: RCI method: validation methodology.

4.2.1 Dataset collection

The employed dataset consists of two months of driving data, logged with the embedded device of three equal-type 18-meter buses driving in a city in the Netherlands. The data contains various routes, logged at 10 Hz from the following sources. Vehicle localization is obtained through the Global Positioning System (GPS) system connected to the CYB device, vehicle data through a CAN-bus connection, and schedule information over an interface with the BISON KV6 trip status server [20]. An overview of the available data points is presented in Table 4.1.

Source	Variable	Unit	Type
GPS	latitude	Degree	float
GPS	longitude	Degree	float
GPS	heading	Degree	float
CAN	Battery Current	Ampère	float
CAN	Battery Voltage	Volt	float
CAN	Drivetrain Current	Ampère	float
CAN	Drivetrain Voltage	Volt	float
CAN	Longitudinal velocity	km/h	int
BISON	Rotation ID	-	int
BISON	Line ID	-	string
BISON	Trip ID	-	int

Table 4.1: Trip dataset variables for RCI validation.

4.2.2 Route filtering

A route is characterized by the unique combination of a Line ID, start location, and stop location. The data for these routes was extracted with the developed Algorithm 1, attached in Appendix F, which automatically extracts trip segments and groups the data per route. The behavior of this algorithm can be summarized as follows. First, it extracts the data segments per unique line-trip combination, and obtains the start and stop names and coordinates from a lookup table. The data segment is then trimmed to start and stop near the right coordinates. If the segment starts and ends within the defined radius from the table, it is saved per route.

4.2.3 Sorting training and validation

Sorting the trip data into a training and validation dataset is the next step in the methodology. This was done with data sorting Algorithm 2, attached in Appendix F. The training data is used to generate the reference trip for each route by calculating the RCI, averaged auxiliary power, and remaining time profile, described in Section 4.1.2. The validation data is the input for the simulations discussed next.

4.2.4 Simulate trip energy estimate

The simulation goal is to minimize the Mean Percentage Error (MPE) and the Standard Deviation (SD) of the Percentage Error (PE), for each point along the route. Additionally, the error at 20% trip progress should be within the range [-25,+25]%, according to NF-1-B. This is calculated with Equation 4.6 to 4.8, where $E_{estimate,i}$ and $E_{measured,i}$ are the estimated and measured total energy consumption for the trip i respectively, s is the distance for which the equation is evaluated, and N is the total number of trips.

$$PE_i(s) = 100 \times \frac{E_{estimate,i}(s) - E_{measured,i}(s)}{E_{measured,i}(s)} \tag{4.6}$$

$$MPE(s) = \frac{1}{N}\sum_{i=1}^{N} PE_i(s) \tag{4.7}$$

$$SD(s) = \sqrt{\frac{\sum_{i=1}^{N}[PE_i(s) - MPE(s)]^2}{N-1}} \tag{4.8}$$

The simulations were done by running Algorithm 3, the pseudo-code can be found in Appendix F. This algorithm automatically simulates the RCI energy estimation method for all trips on a route, and updates the reference trip using (4.5), with a new trip generated after each iteration of the simulation.

4.3 Results of RCI trip simulations

Extracting the trips from the two months of logged data provided approximately 2000 trips containing 16 different routes, with a minimum of 50 trips per route. These trips were sorted into training and validation data sets, with per route 30 trips for training, and at least 20 trips for validation. The simulation results for the remaining 1334 validation trips are presented in Figure 4.3, which shows the mean and standard deviations in separate plots for the total, drivetrain, and auxiliary energy consumption estimate errors. Note here that the y-axis limits are matched, and the auxiliary estimate has a significantly larger error standard deviation. Additionally, the drivetrain error rapidly converges to zero mean error, while the auxiliary estimate approaches zero at the trip's end.

Moreover, the total energy consumption estimate error statistics are used to reflect on the accuracy requirement NF-1-B. This reveals that the mean converges to zero around 10% progress. However, with a 2 SD range of [-22.4,+22.9], the error is close to the limits of [-25,+25]%. Evaluating the dataset reveals that on average, the total energy consumption consists of approximately 20% auxiliary and 80% drivetrain. Thus, the auxiliary error propagates to a lower extent into the total error.

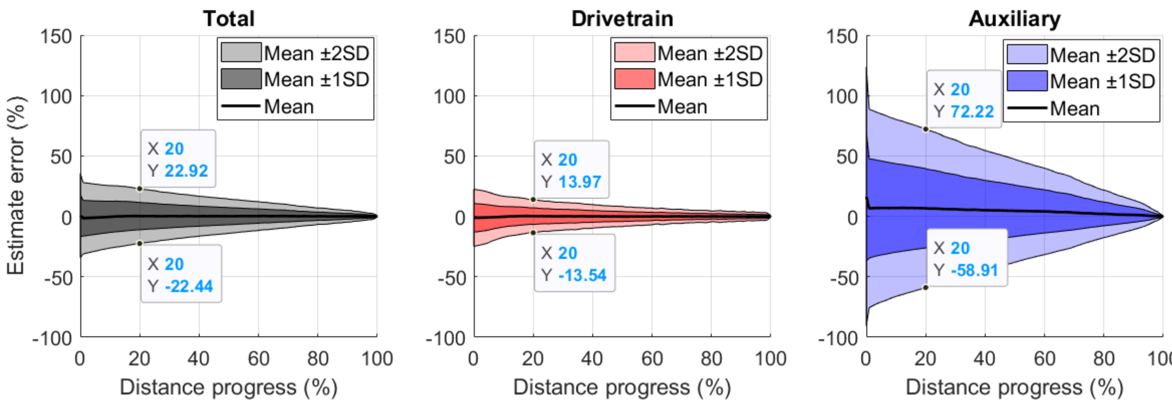
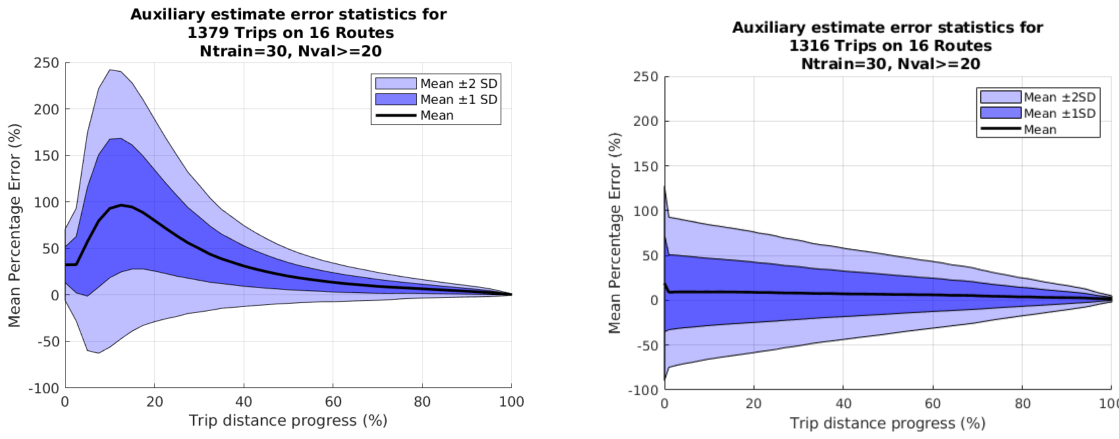


Figure 4.3: RCI method: error of estimation components: total, drivetrain and auxiliary. The error is defined as the Mean Percentage Error of the trip total energy estimate.

Lastly, Figure 4.4 presents the comparison of the auxiliary estimation part of the original method proposed in [1], and the adjusted method outlined in Section 4.1. Note here that both y-axes have equal limits. From these plots, it becomes clear that the proposed auxiliary estimation method reduces the MPE, by eliminating the peak at the start of the trip.



(a) Old method [1] **(b)** Improved method

Figure 4.4: RCI method: comparison of initial and improved auxiliary estimation. The error is defined as the Mean Percentage Error of the trip total energy estimate.

4.4 Discussion

The results of the enhanced RCI method reveal that the total energy consumption estimate error stays within the bounds [-22.4, +22.9] percent error for over 1300 simulated trips, which satisfies the accuracy requirement NF-1-B. The drivetrain part presents promising results, though the auxiliary part is relatively inaccurate.

Since the goal is to realize a model that estimates the total energy consumption, while minimizing both the mean and the standard deviation, the question arises of whether the RCI approach is the best way to accomplish this. Therefore, the next chapter presents the design and evaluation of a Machine Learning (ML) method, with as goal to compare it with the RCI method.

5 Modeling approach 2: Machine Learning

The previous chapter established a performance baseline with the Route Characteristics Indicator (RCI) method. This chapter explores the possibilities of improving the estimation by developing a data-based method with Machine Learning (ML). The development was done according to the CRoss Industry Standard Process for Data Mining (CRISP-DM), described in Appendix G. The process consists of the following steps:

1. **Method design (Section 5.1)**
   (a) High-level design
   (b) Energy consumption prediction equations
   (c) Evaluation criteria
2. **Data preparation (Section 5.2)**
   (a) Extract trip data
   (b) Convert units
   (c) Normalize data (e.g. kWh/km, events/km)
   (d) Generate features
   (e) Filter incomplete entries
   (f) Remove outliers
3. **Modeling (Section 5.3)**
   (a) Feature selection (automatic or manual)
   (b) Standardize and normalize the data
   (c) Splitting the data for training, validation, and testing
   (d) Training the algorithm (grid-search or final training)
4. **Evaluation (Section 5.4-5.5)**
   (a) Metrics evaluation through test set predictions
   (b) Export the model
   (c) Validate through trip simulations

The main steps of this process will be discussed in this chapter.

5.1 Method design

The evaluated high-level design is presented in Figure 5.1, and has two main inputs. The embedded device collects vehicle data, and the BISON KV6 system provides real-time trip status information. More information on the collected data is provided in Section 5.2. The trip status is used for identifying the scheduled route, which is used to calculate the remaining distance. The inputs are sent to a trip data accumulator, which normalizes these values with respect to traveled distance, and forwards these to the drivetrain and auxiliary models. The normalization is done to determine an energy consumption estimate in kilowatt-hours used per kilometer, $\bar{E} = kWh/km$.

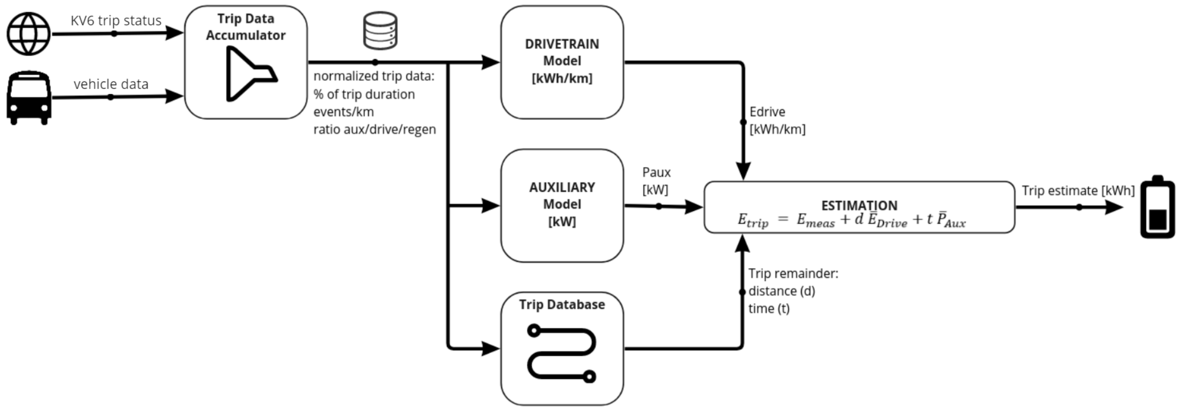


Figure 5.1: Machine Learning model architecture. The schedule and trip data are fed to an accumulator, which normalizes the data and forwards it to the prediction models.

Two main variants were evaluated throughout the project. Equation 5.1 describes the first method, employing two estimators that model the drivetrain and auxiliary parts separately. Equation 5.2 describes the second, which uses a single model for estimating the total energy consumption:

$$E_{trip}(s) = E_{measured}(s) + D_r(s)\bar{E}_{drive}(s) + t_{rem}(s)\bar{P}_{aux}(s) \tag{5.1}$$

$$E_{trip}(s) = E_{measured}(s) + D_r(s)\bar{E}_{total}(s) \tag{5.2}$$

Here, the symbols are defined as follows. E_{trip} is the trip energy consumption estimate, D_r the remaining distance, t_{rem} the remaining time, $\bar{E}_{drive}$ and $\bar{E}_{total}$ the corrected normalized drivetrain and total energy, and $\bar{P}_{aux}$ the corrected auxiliary power estimate.

The model predictions $\bar{E}_{drive}$, $\bar{E}_{total}$, and $\bar{P}_{aux}$ are corrected with the measured values by evaluating (5.3), which is used for all estimates. Here, N_c is the corrected estimate, for either the total, drivetrain, or auxiliary energy consumption. $N_{estimate}$ is the estimate, $N_{measured}$ is the measured normalized energy consumption value, W is the weight calculated with (5.4).

$$N_c(s) = (1 - W(s))N_{estimate} + W(s)N_{measured} \tag{5.3}$$

$$W(s) = \max\Big(0, \frac{p(s) - p_{init}}{1 - p_{init}}\Big), \quad p(s) = \frac{s_{current}}{s_{triptotal}} \tag{5.4}$$

The weight $W(s)$ in the correction can be considered as a measure of confidence in the measured value. Initially, a weight of 0 is used until the progress exceeds the initialization distance threshold p_{init}, from where it linearly approaches 1 at 100% trip completion. The implementation discussed in this thesis used $p_{init} = 0.15$.

Evaluation criteria

The main criteria for selecting a ML model are running time, accuracy, and transparency, on which the existing algorithms score differently. Simple methods such as linear regression require less running time than nonlinear methods and are easier to interpret. The more complex non-linear methods require more running time, are more accurate for non-linear data relations, and are more difficult to interpret. Since the selection of a suitable model is not trivial, the remainder of this chapter evaluates several methods before selecting the most suitable one.

Performance metrics

The main criterion for EVEE is to have an estimate of the highest accuracy, requirement NF-1-B, which shall be updated approximately once per minute, requirement NF-1-A. Therefore, running time is not considered a vital selection criterion and the three applied metrics measure the accuracy of the prediction. First of all, the Mean Absolute Percentage Error (MAPE) is minimized during model training:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_{true,i} - y_{pred,i}}{y_{true,i}}\right|. \tag{5.5}$$

Additionally, the Coefficient of Determination (R2) is evaluated with model predictions after training. The R2 metric is a measure of the goodness of fit of a model [21], and varies from 0 to 1, with a perfect score being 1. It is calculated as follows:

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_{true,i} - y_{pred,i})^2}{\sum_{i=1}^{N}(y_{true,i} - \bar{y}_{pred,i})^2}. \tag{5.6}$$

Lastly, the Root Mean Squared Error (RMSE) in Equation 5.7 is used as a measure for the residual error of the model:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_{true,i} - y_{pred,i})^2}{N}}. \tag{5.7}$$

In Equation 5.5-5.7, $y_{true,i}$ is the real value, $y_{pred,i}$ is the predicted value, and $\bar{y}_{pred,i}$ is the mean of the predictions.

5.2 Data preparation

The accuracy of a Machine Learning (ML) model is affected significantly by the amount and quality of the data used for training the model. The collection of a new dataset is outside of the scope of this project, due to the limited time available for doing so. Thus, two prior available datasets are evaluated, which have been recorded by 42 equal-type 18-meter buses, driving in the same city as described in Chapter 4. Some data characteristics are listed in Table 5.1.

Name	Data point count	Start date	End date	Duration
Driver segments	112.000	June '19	Dec '21	2.5 yr
Trip segments	56.000	Jan '21	Dec '21	1 yr

Table 5.1: Datasets evaluated during machine learning model development.

These datasets have a slightly different representation of trips, since the **driver segments** dataset contains blocks of trips, which are grouped per driver, and the **trip segments** data consists of vehicle trips, which are reset when the ignition switches off/on. The remainder of this report presents the results of the trip segments dataset only, which resulted in the best prediction accuracy.

Available data points

A complete overview of the available signals with their description can be found in Appendix I, and can be summarized as follows:

- **Energy:** total, auxiliary, drivetrain, regenerated, battery SoC at start and stop
- **Operations:** distance, duration, idle time, rolling time, driving time, ambient temperature
- **Bands:** speed, acceleration, accelerator pedal, motor rpm, engine load, battery load
- **Driving events:** hard & excessive acceleration and braking, full throttle, mechanical braking
- **Drivestyle feedback display:** time that display lights up green, yellow, and red
- **GPS start and stop:** latitude, longitude, heading, stop name
- **Trip info:** rotation ID, trip ID, line ID

Feature generation

Feature generation considers the augmentation of the data to realize additional variables for model development, and was done by combining data points into e.g. a Driving Score, and performing mathematical operations on the data. These operations aim to transform the data, by for example linearization, improving suitability for a linear regression model.

5.3 Model development

The nature of the prediction problem fits in the class of regression methods. Hence, this section discusses the development of such a model, which was realized using a Python-based Jupyter notebook.

Feature selection

During feature selection, the aim is to select variables that best fit the model, thereby maximizing its accuracy. This was done by selecting variables that have a high correlation with the target variable: normalized energy consumption. Initially, this was done through Pearson correlation analysis, but since this only identifies linear correlations, an additional analysis was performed by pair-wise inspecting the data points to identify nonlinear relations, such as the effect of ambient temperature. This analysis is discussed in detail in Appendix I.

In practice, feature selection proved to be an iterative process, starting with model training, followed by trip simulations, resulting in a change of training features. Experiments reveal that using more parameters during training results in a higher training accuracy. However, trip simulations are affected negatively due to volatility in the estimates. The root cause investigation presented in Appendix H exposes the data accumulator as the cause for this. Thus, feature selection included finding parameters with little variation from the accumulator, while having a high correlation with the model.

Algorithm selection and optimization

The selection and optimization of a modeling method has been achieved through a grid-search optimization that minimized Equation 5.5-5.6. The methods included linear regression up to non-linear Neural Networks. The grid-search results are presented in Table 5.2. Of all methods, the Neural Network architecture demonstrated the highest accuracy and was selected as the primary modeling method. Since this method can be considered a black box, a sensitivity analysis was required to identify the dominant parameters in the estimate, which is further discussed in Appendix I.

	MAPE	**R2**	**RMSE**
Method	[%]	[-]	[kWh/km]
Multiple Linear Regression	9.2	0.49	0.018
Ridge	9.2	0.49	0.018
Lasso	10.5	0.38	0.022
ElasticNet	9.4	0.49	0.018
Decision Tree	8.1	0.56	0.016
Random Forest	7.6	0.62	0.013
Gradient Boosted Tree	4.5	0.84	0.006
Neural Network: Multi-Layer Perceptron	4.0	0.84	0.006

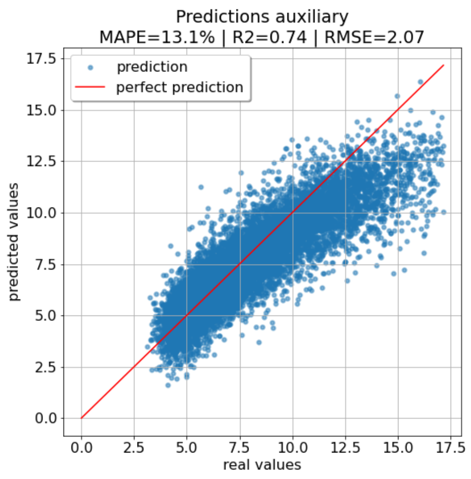
Table 5.2: Machine Learning algorithm grid-search results. Optimized for target energy kWh/km. 30 features are included which have a correlation of over 0.05 with the total energy consumption. The Neural Network architecture consisted of three hidden layers with 30 nodes each and a ReLU activation function, with a single output node with linear activation.

5.4 Training results

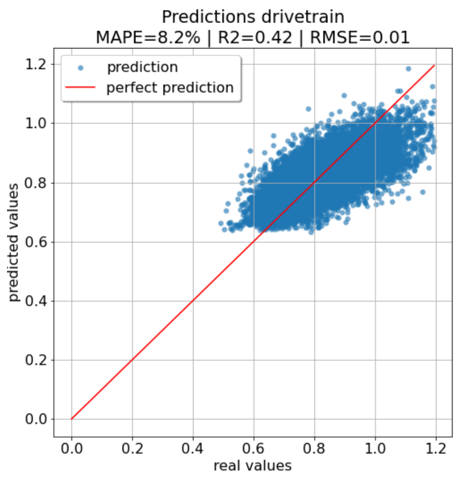
The Neural Network architecture consisted of three hidden layers with 30 nodes each and a ReLU activation function, with a single output node with linear activation. The training results for the total, auxiliary, and drivetrain models are presented in Figure 5.2, which have been developed with a train/validation/test data size ratio of 80/16/4%. The validation data has been used to terminate the training process once overfitting had been detected. The models use the following inputs:

- **Ambient temperature:** T_a, T_a^2, $(T_a - 20)^2$.
- **Date:** Month of the year, day of the week, periodical features, e.g. cos(MoY), sin(MoY).
- **Time:** hour of the day, periodically, periodical features cos(HoD), sin(HoD), and a rush hour boolean with rush hours between 6:30-9:30 and 15:30-19:00.
- **Drive Score**: a measure of driver aggressiveness, established by evaluating the relative time that the driver spends in defined acceleration bands. $\text{DriveScore} = \sum_{i=1}^{N} W_i \times (T_i - T_{ref,i})$, where T_i is the relative time in band i, and $T_{ref,i}$ is a target value.
- **Route identifier**: a unique combination of the lineID, and start and stop names
- **Energy ratios, used in the auxiliary model only**: ratio of measured energy auxiliary/total, drivetrain/total, and regenerated/total.

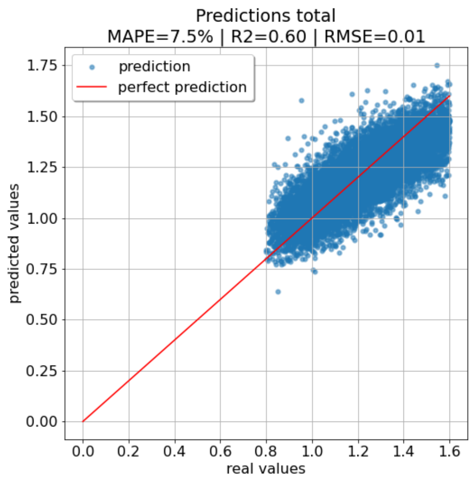
Data analysis shows that the energy ratios have a strong correlation with the energy estimate. However, a drivetrain model trained on this performs insufficiently due to volatility in the variables, see Appendix H. The auxiliary model is less sensitive to this and therefore uses these variables as inputs.



(a) Auxiliary power (kW)



(b) Drivetrain energy normalized (kWh/km)



(c) Total energy normalized (kWh/km)

Figure 5.2: Machine Learning model predictions for the auxiliary, drivetrain, and total energy models.

5.5 Trip data simulation

The trip simulations, which are used for verifying the accuracy requirement, are realized similar to the RCI validation methodology to allow a fair comparison of the results. The simulations have been done with newly collected data, since the RCI data lacked several required parameters, and the training data lacked intermediate data points. The new data was collected by two buses, sampled at 1 Hz, containing the accumulator variables, and the BISON KV6 trip status.

Due to time constraints, 113 trips were collected, compared to approximately 2000 trips for the RCI method in Section 4.2. Experiments revealed that the lowest MPE was achieved for the split model, consisting of a drivetrain and auxiliary estimator. The results of this model are presented in Figure 5.3. Note here, that during training only the MAPE metric was available, but that the trip simulations evaluate the MPE to allow the calculation of the standard deviation.

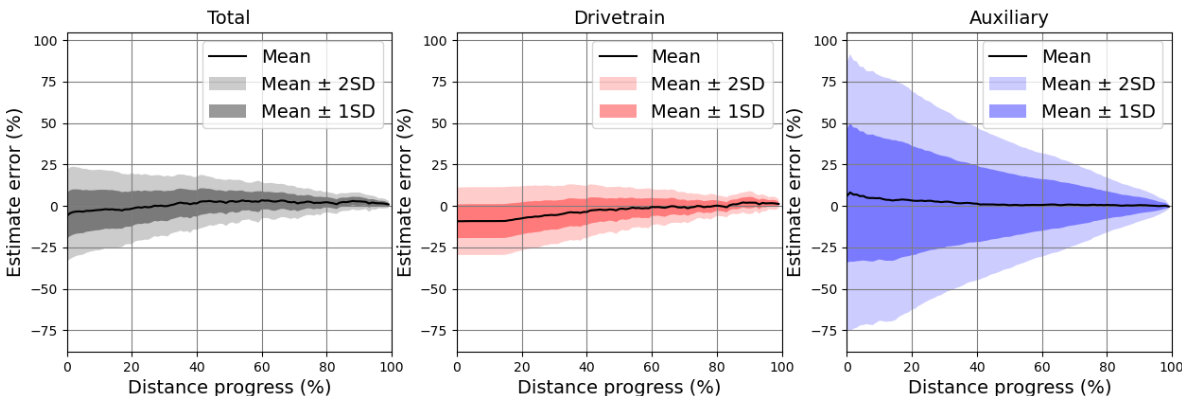


Figure 5.3: Machine Learning model trip simulations. The Mean Percentage Error for 113 simulated trips.

5.6 Discussion

The results presented in Figure 5.3 show that the MPE converges to zero rapidly, and the total estimation error stays within the range [-23,+19.8]% for 95% of 113 simulated trips. Moreover, the auxiliary estimation has a considerably high standard deviation, indicating the Machine Learning model is relatively inaccurate, similar to the RCI method.

Two models have been developed for estimating the energy consumption while driving, based on the RCI and Machine Learning methods. Therefore, the next chapter selects a method as the foundation for the final design EVEE.

6 Design of the EVEE system

In the previous chapters, the problem analysis has been presented and several methods for trip energy consumption estimation have been evaluated. This chapter builds on this by presenting the second part of the development methodology, see Figure 6.1. First an energy consumption estimation modeling method is selected in Section 6.1, followed by a selection of a software deployment concept in Section 6.2. Moreover, based on these decisions a design is presented for the software in Section 6.3-6.5. All design figures presented in this chapter are according to the Systems Modeling Language (SysML).

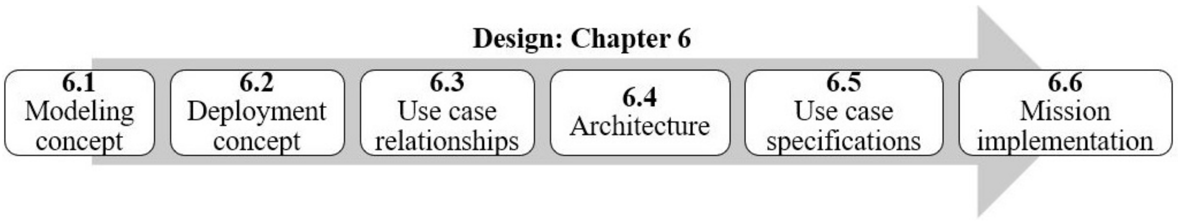

Design: Chapter 6
6.1 Modeling concept
6.2 Deployment concept
6.3 Use case relationships
6.4 Architecture
6.5 Use case specifications
6.6 Mission implementation

Figure 6.1: EVEE system development methodology: design phase

6.1 Modeling concept selection: RCI versus Machine Learning

The selection of an energy estimation method was done with the non-functional requirement NF-1-B, which states that the estimate should have less than 25% MPE, after 20% of the traveled trip distance, for at least 95% of at least 500 trips. Therefore, a 95% confidence interval is established with the mean error and two standard deviations (SD). The results of the simulations described in Chapter 4 and 5 are summarized in Table 6.1.

From comparing the total estimate with MPE ± 2SD it seems that both algorithms have similar performance, and both satisfy requirement NF-1-B. However, the standard deviation of the RCI method is 1% higher than the ML method. This difference can be explained by the fact that the evaluated number of trips for the ML model is significantly lower. Hence, the RCI model is expected to perform better while it seems to have a slightly higher error at 20%.

Furthermore, the RCI method appears to have a lower error in the drivetrain part, which can be explained by the detailed trip modeling with the drive-cycle-like approach. In contrast, the ML method uses high-level trip characteristics only. Moreover, the ML method seems to have a lower error in the auxiliary part. Concluding, the RCI method was selected for the initial version of EVEE, since it performs better on the drivetrain estimate, which is often approximately 80% of the total energy consumption estimate.

Model	Estimation	Nr trips	MPE ± 1SD at 0% trip progress	MPE ± 1SD at 20% trip progress	MPE ± 1SD at 40% trip progress	MPE ± 2SD at 20% trip progress
RCI	Total	1334	-0.5 ± 14.8	**0.2 ± 11.3**	0.2 ± 8.5	**[-22.4, 22.8]**
ML	Total	113	-6.1 ± 13.8	**-1.6 ± 10.7**	1.6 ± 8.0	**[-23, 19.8]**
RCI	Drivetrain	1334	-0.5 ± 12.1	0.2 ± 6.9	0.0 ± 5.1	-
ML	Drivetrain	113	-9.2 ± 10.2	-7.5 ± 9.7	-3.6 ± 7.9	-
RCI	Auxiliary	1334	16.5 ± 53.6	6.6 ± 32.8	5.1 ± 24.7	-
ML	Auxiliary	113	6.1 ± 40.3	3.3 ± 32.9	1.4 ± 23.1	-

Table 6.1: Comparison of Machine Learning and RCI energy estimation models. Mean Percentage Error (MPE) at 0, 20, and 40 percent trip distance progress.

6.2 Deployment concept selection: device versus cloud

The selection of a software deployment concept is vital to the operation of EVEE, and therefore three concepts were compared. These include deploying on the OWA3X embedded device, on the OWA450 and OWA3X devices, and as a cloud service. The comparison of these concepts was done with the following decision variables:

1. **Cost of data transfer** is the dominant cost of the product and constrains the reduction in operational costs, as well as indirectly influencing the profit margin. This is specified in requirement NF-2-A.
2. **Deployability** is important for the product to have a useful impact on the current operations. The software should be useable with as many buses as possible (NF-7-A).
3. **Trip initialization time** is characterized by the delay between the start of a trip and the energy consumption estimation. The RCI method needs a reference trip before the estimation can start. These trips need to be stored in the cloud, since logging individually would lead to initialization time before the software can provide the first estimate. Hence, the time required for downloading the trip has an impact on the delay before starting estimation (NF-1-D).
4. **Operational risk** should be minimized. The software stack currently deployed on the embedded devices is a stable release, and making any additions to this stack needs extensive testing. Inherently, making changes to this software could lead to instability in the operation (NC-1-C).
5. **Storage resources** should be considered especially for the OWA3x, which currently has only 10MB of storage available.
6. **Computational resources** are again limited especially on the OWA3x device.
7. **Implementation effort** is important to consider given the limited time for project completion.

A selection matrix including all concepts and decision variables is presented in Table 6.2. Additional information on the concept selection, such as hardware specifications, and cost calculations of the data transfer can be found in Appendix J. In the presented matrix the cloud Service concept scores best overall and was therefore selected for the software implementation.

		Deployment concept		
Decision variable	**Weight factor**	**OWA3x**	**OWA450**	**Cloud Service**
Costs of data transfer	3	1	1	3
Deployability	3	3	1	5
Trip initialization time	2	1	3	5
Operational risk	2	1	1	3
Storage resources	2	1	3	5
Computational resources	1	1	3	5
Implementation effort	3	5	5	1
	Total:	34	38	**58**

Table 6.2: Software deployment concept selection matrix. Values range from 1 (poor) to 5 (good).

6.3 Use case relationships

The previous sections present the selection of the RCI modeling method, combined with a cloud-based software deployment concept. Next, the desired functionality that has been identified through the mission analysis (see Section 2.6), is further specified by defining relationships between use cases and actors. For indicative purposes, the use cases of existing ZE services are presented in Figure 6.2. In addition, the desired use cases implemented by EVEE are presented in Figure 6.3. In both figures, use cases denoted green have been realized prior in the Zero Emission services and embedded devices, white use cases are implemented by the presented EVEE design, and red use cases are recommended for future implementations. These were not realized due to the prolonged development time for realizing the interface.

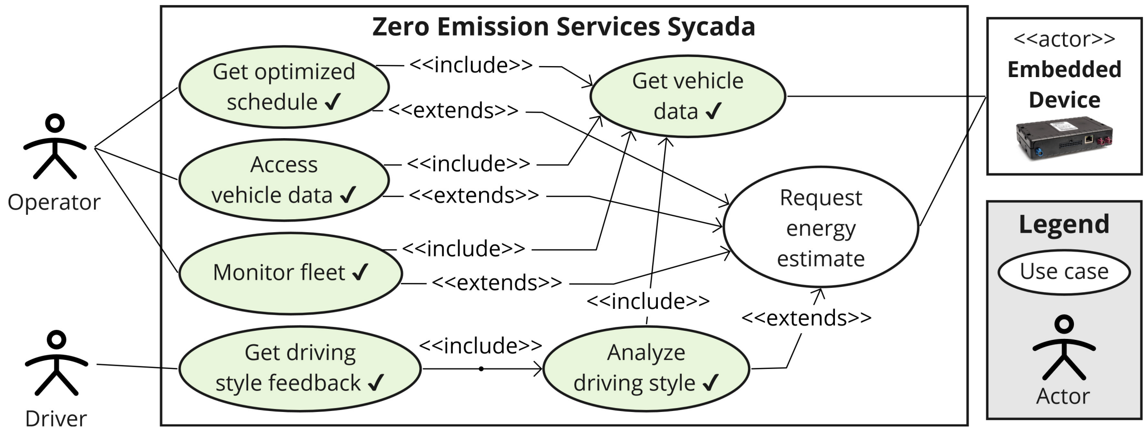


Figure 6.2: Zero Emission Services Use Cases (SysML). Indicative by presenting the relationship between existing services and the EVEE system to be developed. Green use cases, denoted with a check mark, were prior realized in the Zero Emission cloud services and the embedded device, and are therefore not implemented by EVEE. White use cases are implemented by the presented EVEE design.

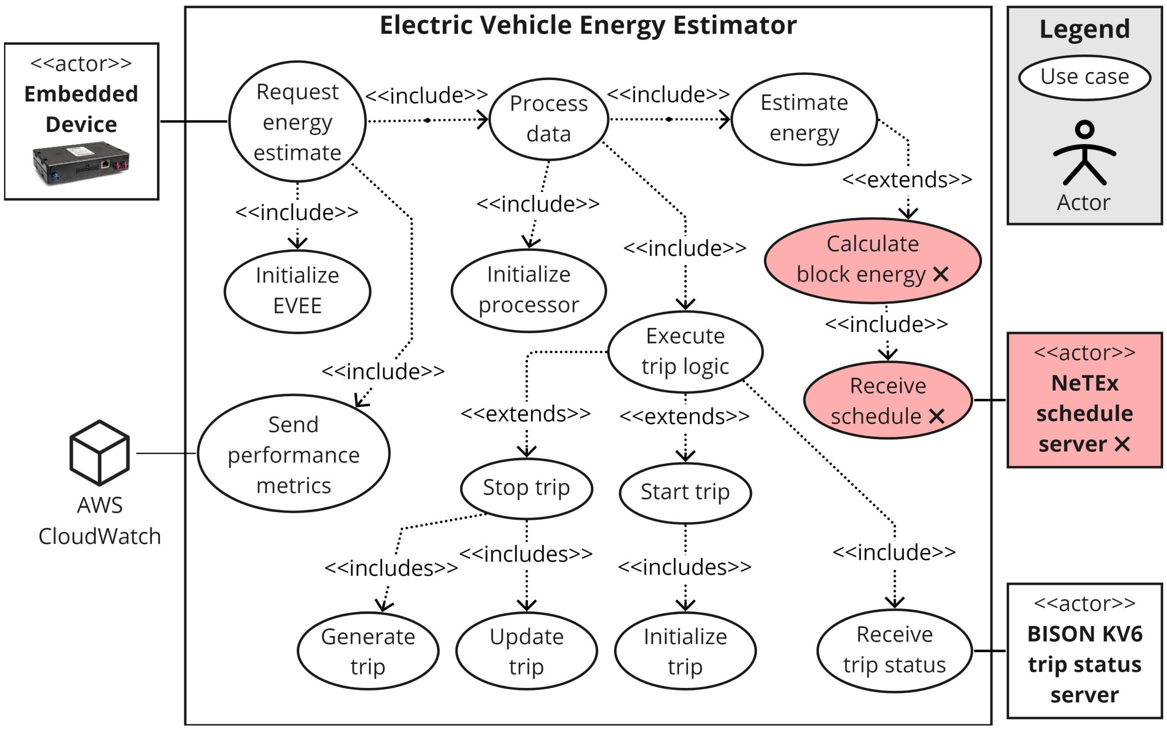


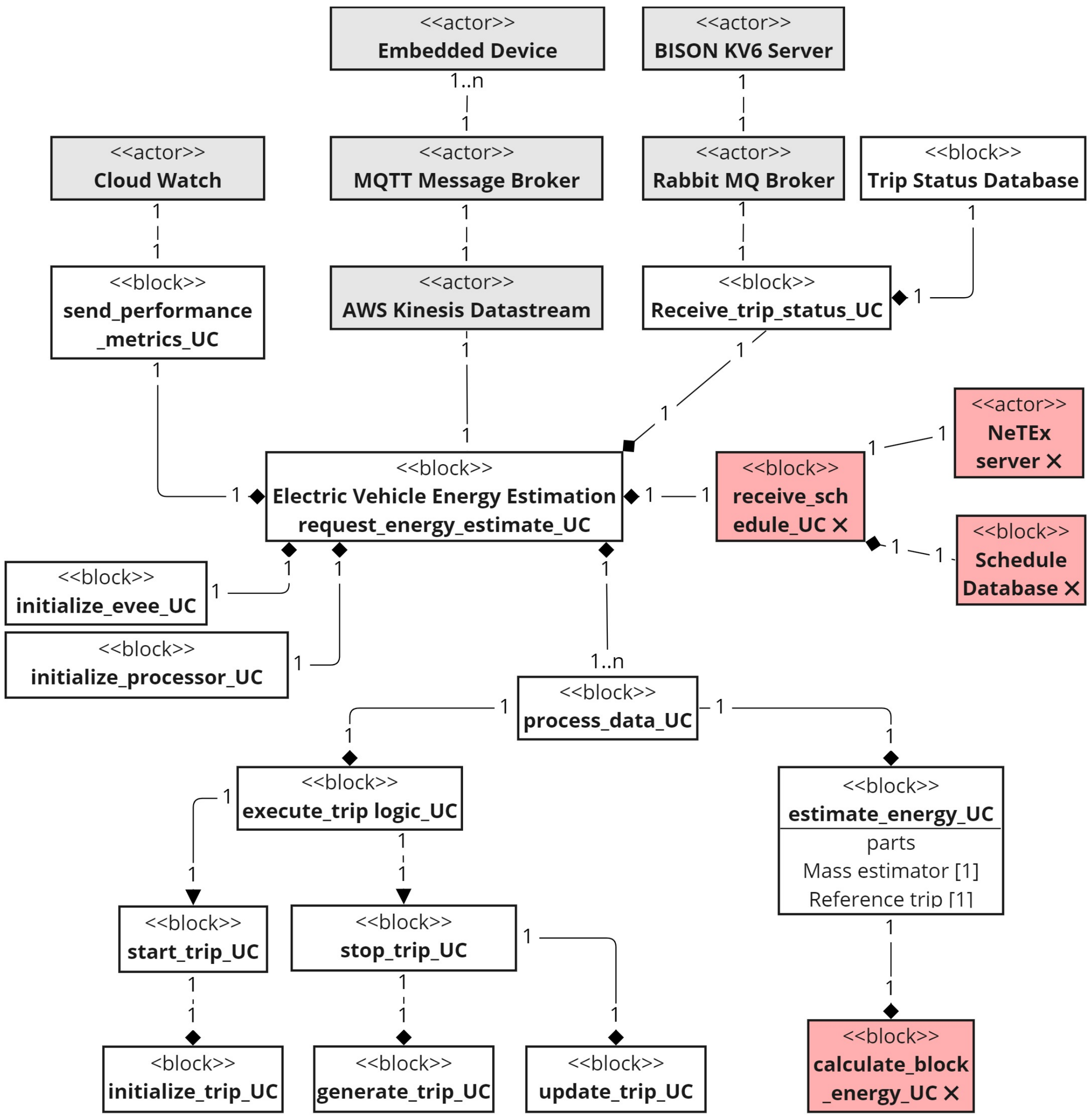
Figure 6.3: EVEE system use case diagrams (SysML). The white use cases are implemented by the current version of EVEE, and those colored red, denoted with an X, are future work.

6.4 Architecture

The architecture diagram in Figure 6.4 presents the components of the software that implement the use cases described by Table 2.4 and Figure 6.3. The *embedded device* requests an energy consumption estimate from the EVEE service, which responds with the estimate. The architecture includes recommended future work, which extends the trip energy consumption estimation to a block of trips. This requires extending the software with an interface to the NeTEx server.

The communication works as follows. The *embedded devices* post the request on the *MQTT Message Broker*, which distributes the message to anyone listening to the topic name of the message. The *AWS Kinesis Datastream* listens to these topics, collects the messages, and stores them in a datastream, which can be described as a message queue. Using this datastream allows the collection of all the messages, even when EVEE is temporarily not available. For example, when a software update is released, or many messages arrive at the same, exceeding the throughput limit. EVEE processes the messages with the First In First Out (FIFO) principle, to ensure messages processing in chronological order. Additionally, this allows scaling of the software by connecting multiple services to the stream, thereby addressing the scalability requirement NF-7-A.

The service initializes a concurrent processor per embedded device, allowing the processing of requests in parallel. The processors execute the trip start/stop logic, using the trip status messages stored in the trip status database, which are obtained from the *BISON KV6 server*. After executing the trip logic, the estimator updates the energy consumption estimate, which requires a *mass estimator* and a *reference trip*. Lastly, the performance metrics are then sent to the *AWS Cloud Watch platform*.



<<actor>> Embedded Device
<<actor>> BISON KV6 Server
<<actor>> Cloud Watch
<<actor>> MQTT Message Broker
<<actor>> Rabbit MQ Broker
<<block>> Trip Status Database
<<block>> send_performance _metrics_UC
<<actor>> AWS Kinesis Datastream
<<block>> Receive_trip_status_UC
<<actor>> NeTEx server ✕
<<block>> Electric Vehicle Energy Estimation request_energy_estimate_UC
<<block>> receive_schedule_UC ✕
<<block>> Schedule Database ✕
<<block>> initialize_evee_UC
<<block>> initialize_processor_UC
<<block>> process_data_UC
<<block>> execute_trip logic_UC
<<block>> estimate_energy_UC
parts
Mass estimator [1]
Reference trip [1]
<<block>> start_trip_UC
<<block>> stop_trip_UC
<block>> initialize_trip_UC
<<block>> generate_trip_UC
<<block>> update_trip_UC
<<block>> calculate_block _energy_UC ✕
1..n
1

Figure 6.4: EVEE system architecture diagram (SysML), presenting EVEE's main components, the relationships of actors to these components, and what use cases (ellipses) are implemented in these components. Use cases in red (denoted with an X) are recommended for future work.

6.5 Use case specifications: system behavior

The defined use cases are specified with the activity diagrams presented in this section, which model the behavior of the system. The core functionality of EVEE is presented in Figure 6.5, and the trip logic is shown in Figure 6.6. Each diagram has a title and a use case specification, indicating which use case is implemented. The colors indicate the coupling between diagrams. For example, the activity 'Start request consumer' runs the 'Start request consumer' diagram.

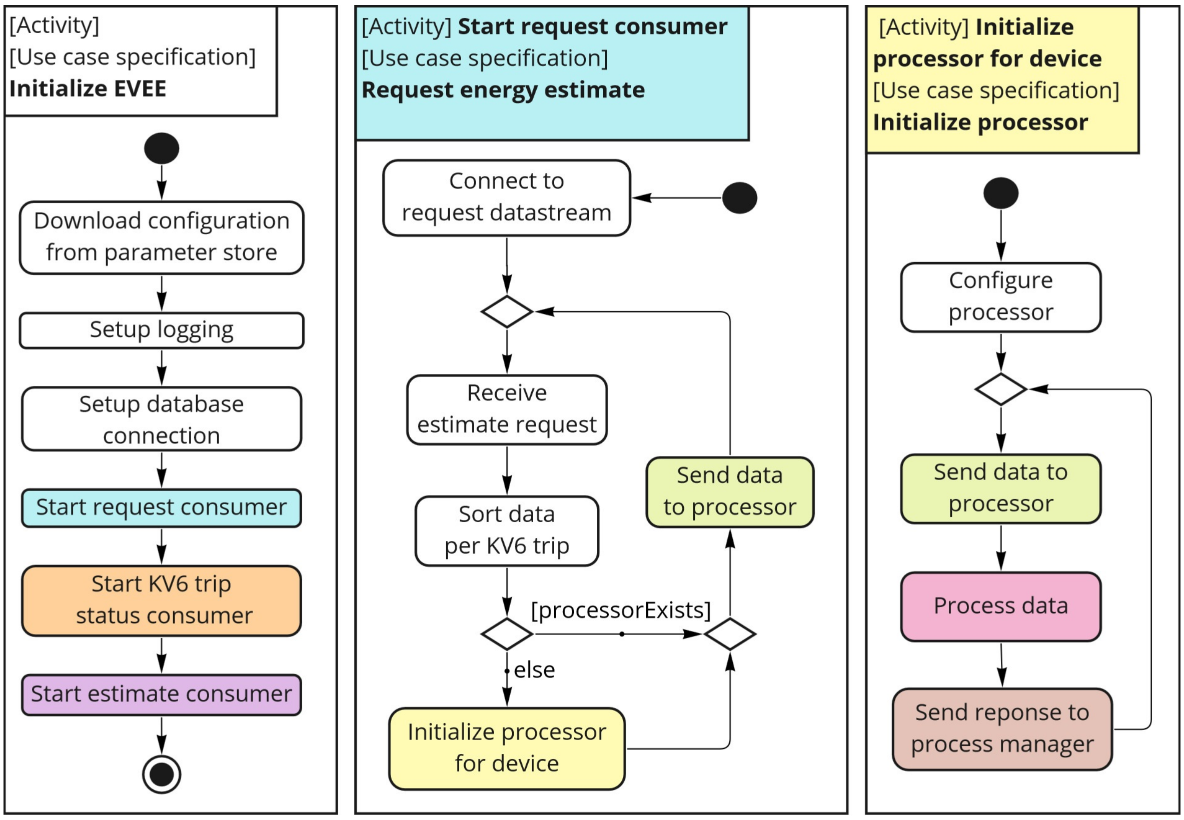


Figure 6.5: EVEE system behavior 1: use case specification diagrams (SysML). Colors denote a coupling between activities and diagrams. The diagrams for 'start KV6 trip status consumer', 'start estimate consumer', and 'process data' are specified in Figure 6.6.

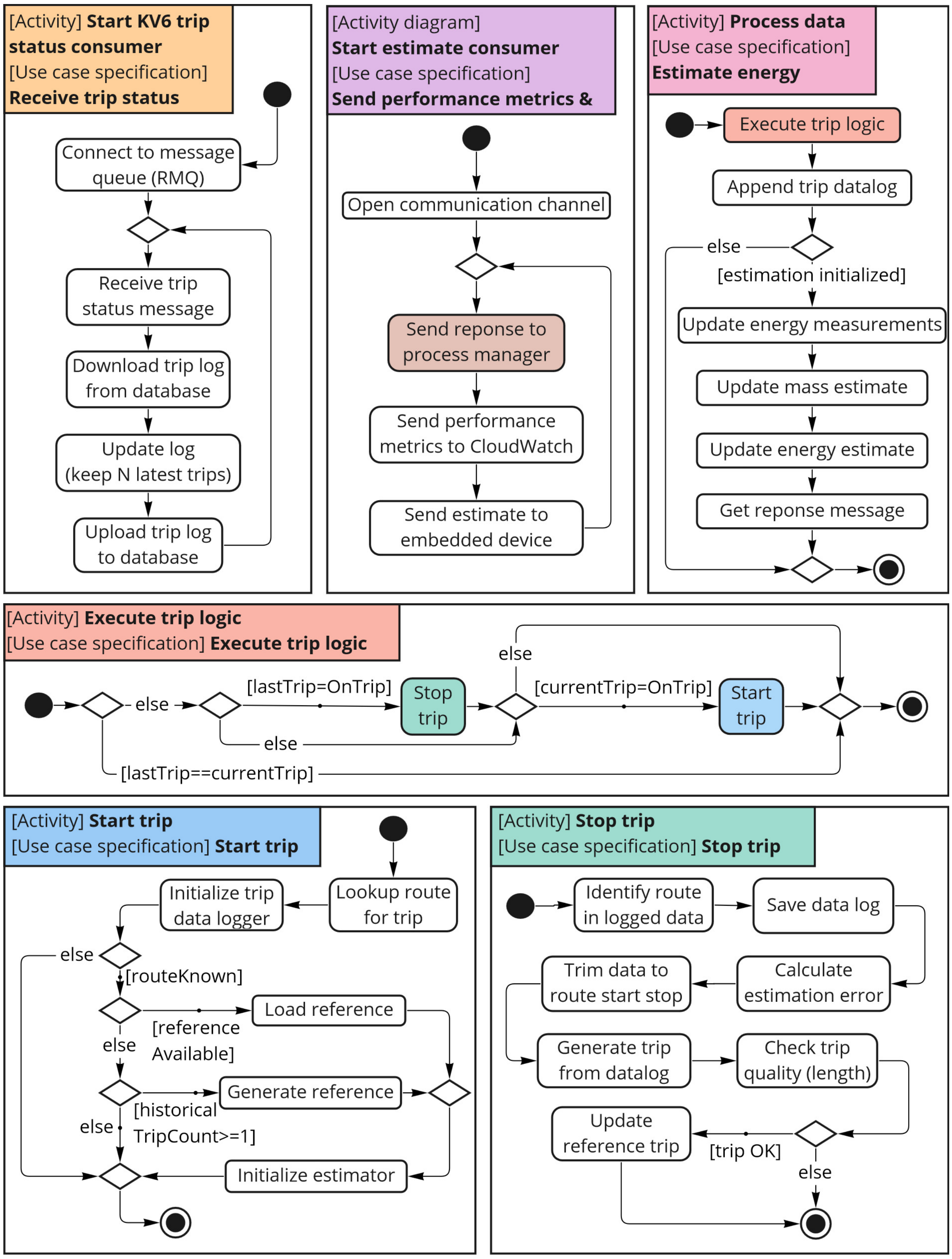

[Activity] Start KV6 trip status consumer
[Use case specification] Receive trip status
Connect to message queue (RMQ)
Receive trip status message
Download trip log from database
Update log (keep N latest trips)
Upload trip log to database
[Activity diagram] Start estimate consumer
[Use case specification] Send performance metrics &
Open communication channel
Send reponse to process manager
Send performance metrics to CloudWatch
Send estimate to embedded device
[Activity] Process data
[Use case specification] Estimate energy
Execute trip logic
Append trip datalog
else
[estimation initialized]
Update energy measurements
Update mass estimate
Update energy estimate
Get reponse message
[Activity] Execute trip logic
[Use case specification] Execute trip logic
else
[lastTrip=OnTrip]
Stop trip
else
else
[currentTrip=OnTrip]
Start trip
[lastTrip==currentTrip]
[Activity] Start trip
[Use case specification] Start trip
Lookup route for trip
Initialize trip data logger
else
[routeKnown]
Load reference
[reference Available]
else
Generate reference
[historical TripCount>=1]
else
Initialize estimator
[Activity] Stop trip
[Use case specification] Stop trip
Identify route in logged data
Save data log
Trim data to route start stop
Calculate estimation error
Generate trip from datalog
Check trip quality (length)
Update reference trip
[trip OK]
else

Figure 6.6: EVEE system behavior 2: use case specification diagrams (SysML). Colors denote a coupling between activities and diagrams.

6.6 Mission implementation

The previous sections present all the ingredients for realizing EVEE's mission, introduced in Section 2.6, which is to calculate the trip energy consumption estimate and send this to Sycada's Zero Emission cloud services. An overview of the realized implementation is presented in Figure 6.7.

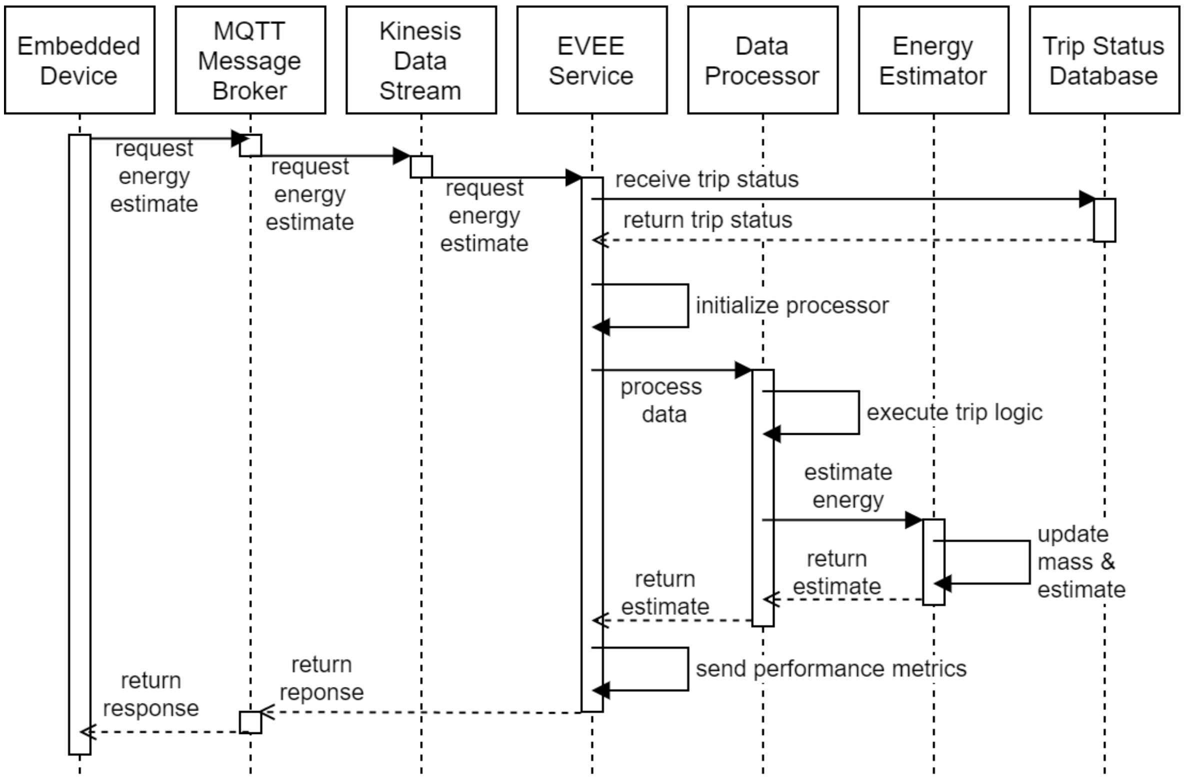


Figure 6.7: EVEE mission implementation: a SysML sequence diagram combining the architecture and the behavior for realizing the mission.

This figure visualizes the message flow beginning with the embedded sending a request, which is forwarded all the way through the EVEE system up to the estimator, from where a response message is generated which is returned back to the embedded device. Since the embedded device already possesses communication with the cloud services, this is not implemented by the EVEE system.

6.7 Discussion

This chapter presents the final design of the EVEE software, which is based on the RCI energy consumption modeling method, and combined with a cloud-based software deployment concept. All ingredients for realizing the mission were introduced, and the implementation of the mission was visualized. Having presented the implementation of the mission, the question arises whether it satisfies the requirements defined in Table 2.2 and 2.3. This will be discussed next in Chapter 7.

7 Validation

The final step in the system development methodology is validating the implementation of the design presented in Chapter 6, through system-level testing of the software during operation.

The EVEE design was implemented in GoLang, and deployed as a service on an AWS web server. The validation was done through experiments with the same 42 buses as discussed the RCI in Chapter 4 and ML in Chapter 5.

The validation focuses on the nonfunctional requirements since these include the specification of the functional requirements. A summary of the validation results is presented in Table 7.1, and additional materials supporting the validation can be found in Appendix L.

This section presents a detailed validation of NF-1-B, which states that the system should estimate the total trip energy, with less than 25% Mean Percentage Error (MPE), after 20% of the traveled trip distance, for at least 95% of at least 500 trips. The estimation error of the final implementation is presented in Figure 7.1. The total estimation error should be within the range $[-25, 25]$ for the mean plus-minus two standard deviations to satisfy this requirement. As the Figure indicates, the total estimate error verifies this requirement and has a negative bias of -3%, which is still considered to be acceptable.

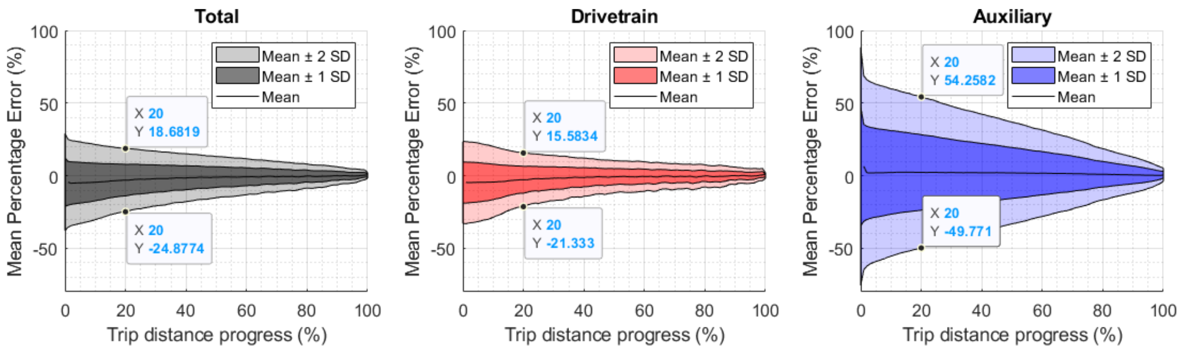


Figure 7.1: Pilot test estimation error: total, drivetrain, and auxiliary components. In total 3680 complete trips were evaluated.

To conclude, the majority of the requirements are satisfied according to Table 7.1, and the up-time requirement NF-1-C is nearly satisfied. Since the proposed customer value is to reduce the risk in the fleet operation, it is vital to have an estimate that meets all the requirements. Therefore, Chapter 8 presents the main conclusions and recommendations for improving and validating the software.

Table 7.1: Validation of nonfunctional requirements through the results of a pilot test. Supporting material is available in Appendix L for validations that are marked with *.

ID	Requirement	Priority	Measurement	Status
NF-1-A	The EVEE system must publish the energy estimate every 60 ± 1.0 seconds while having a maximum standard deviation of 3.0 s measured for a single bus over one day.	Must	mean(Δt) = 60.0 s SD(Δt) = 2.5 s	Verified *
NF-1-B	The EVEE system should estimate the total trip energy, with less than 25% Mean Percentage Error (MPE), after 20% of the traveled trip distance, for at least 95% of at least 500 trips.	Should	Mean error of 3680 trips: -3%, with 95% of trips within: $E = [-24.9, 18.7]\%$	Verified
NF-1-C	The EVEE system should provide an energy estimate at least 99% of the time while on a schedule.	Should	94.2% average over 15 days	Failed *
NF-1-D	The EVEE system should provide its first energy estimate within two weeks after a schedule revision.	Should	route initialization time of 1 week	Verified
NF-2-A	The total costs of operation of EVEE must not exceed €5 per bus per month.	Must	Data: €1.03/month	Verified *
NF-2-B	The bandwidth usage of the EVEE system must be less than 0.05 kilobytes per second per bus, averaged over a minimum of 1 week.	Must	16 day average 0.013 kB/s	Verified
NF-3-A	Initialization and configuration of the EVEE system should take no longer than 30 seconds after power-up.	Must	N_{tests} = 11 t_{init} = 5.0 s	Verified
NF-4-A	The EVEE system should publish the estimation RMSE at least once per trip.	Should	Done	Verified
NF-5-A	The EVEE system should be programmed in GoLang.	Should	Done	Verified
NF-6-A	The EVEE system must comply with the General Data Protection Regulation of the European Union.	Must	Done	Verified
NF-7-A	The EVEE system should be compatible with all OWA3x and OWA450 devices of Sycada.	Must	Done	Verified

8 Conclusions and recommendations

The transition from conventional fossil fuel towards battery electric buses introduces additional risk in the operation through the reduced range and less predictable energy consumption. This risk can be produced by providing insight into the real-time operation of the bus through an online monitoring platform. Project Destination Zero had as its main goal to provide an energy consumption forecast, thereby improving predictability and reducing operational risks. This chapter presents the main conclusions and proposes directions for future work.

8.1 Conclusions

Two methods for modeling the trip energy consumption have been evaluated by performing simulations with historical trip data. The modeling was done by splitting the total energy consumption into a drivetrain and an auxiliary energy consumption component, where the average ratio of drivetrain/auxiliary energy is 80% to 20%. From both methods, the Route Characteristics Indicator (RCI) demonstrates a slightly higher accuracy in the drivetrain estimate, and the Machine Learning (ML) method has slightly higher accuracy in the auxiliary estimate. However, both methods have a relatively large estimation error in the auxiliary component. Additionally, the RCI method demonstrates better performance for real-time prediction during trips, and the ML method for high-level prediction, which ignores detailed trip information.

Based on these results, the final design of the energy estimation software employs the RCI modeling method, deploys as a web service, and receives data from the embedded device in the bus. The proposed method utilizes the repetitive and predictive nature of the trips. The design has been implemented and validated, after collecting experimental data with 42 buses. Analysis of 3680 recorded trips provides the following results:

- The mean estimation error at 20% trip progress was -3% for 3680 trips
- The error after 20% of the trip has been completed stays within the range $[-24.9, +18.7]\%$, for 95% of 3680 trips
- The estimation was done 94% of the time while driving on a schedule
- The total data transfer cost was established at €1.03 per bus per month.

To summarize, the implementation's validation passes on 10 out of 11 requirements. The up-time did not reach the target of 99%. Nevertheless, the pilot experiments demonstrated that the Electric Vehicle Energy Estimation (EVEE) software can improve the predictability of energy consumption, and consequently, it could reduce the risk for the operator. Therefore, the recommendation is to continue developing the software by addressing the activities proposed in the next section.

8.2 Recommendations for future work

To improve the performance of the software, and to satisfy all requirements in a future version, the following activities are proposed:

- **Further improving the auxiliary estimate.** The proposed adjustments of the auxiliary estimate in this report, reduce the error, especially at the start of a trip. However, there are still opportunities to further reduce the error of this part. A possibility is to realize a machine learning model for predicting the mean auxiliary power.
- **Implement NeTEx schedule interface.** The NeTEx interface could not be developed in time for this project. Hence, a workaround was established that identifies the route for each Line-Trip combination. This is identified as an issue that causes the up-time requirement to be invalidated. For enhancing the up-time, the temporary workaround should be replaced with the schedule implementation. Moreover, the schedule is required for realizing a block estimation.
- **Extending the energy estimate from trip to block estimation.** Currently, energy forecasting is done for individual trips of up to approximately 30 minutes. Realizing an estimate for a block of trips can enlarge the prediction horizon to the point that the bus has returned to the depot. This allows it to predict the remaining charge at the end of service, thereby enhancing the usability of the estimate. Moreover, the auxiliary power fluctuates significantly within trips, making short-term estimation difficult. Possibly this fluctuation has less impact on the estimate when the prediction horizon is longer.
- **Maintaining reference trips per time of the day.** A single reference trip is presently maintained per route, which represents the average of many trips throughout the day. By maintaining a reference trip per trip ID, which is driven once per schedule, the reference trip becomes dependent on the time of day. Therefore the effect of external factors such as rush hour traffic, and temperature fluctuations throughout the day can be included.
- **Scaling the drivetrain RCI profile.** The moving average reference trip updating strategy allows adapting to changing circumstances such as seasonal influences. To improve short-term prediction, a high-level prediction model could be established that adapts the RCI profile based on parameters such as the weather forecast, time of day, driving style, etc.
- **Realizing a continuous real-time correction.** Currently, the Kalman filter re-initializes at the start of every trip. To avoid the re-initialization time, it can be worthwhile to investigate the effect of maintaining the Kalman filter's state between consecutive trips.
- **Scaling the software.** The pilot test of EVEE included 42 buses of the same model. Once the remaining requirement is satisfied, the software shall be used with different buses with varying properties and thus different configurations of EVEE, for example, a different initial mass of the reference trip. The automatic configuration for different bus types needs to be realized.
- **Validating the value proposition.** The system requirements have nearly been satisfied, with an accuracy that stays within the defined limits. The proposed customer value, of reducing costs and risks of the operation, should be validated by applying the software in practice. This can include validating the proposed savings by optimizing the planning, by reducing the battery capacity buffers in the schedule.

Bibliography

[1] Dhruv Jagga. Cloud your bus: real-time energy consumption prediction for electric city buses. *Eindhoven University of Technology Research Repository*, (PDEng Report 2022/045), 10 2020.

[2] European Commission. A European Strategy for Low-Emission Mobility. *European Environment Agency*, (Technical Report), 4 2016.

[3] Sustainable Bus. The pandemic doesn't stop the European e-bus market: +22% in 2020. *https://www.sustainable-bus.com/news/europe-electric-bus-market-2020-covid/ (accessed Apr. 4, 2022)*, 4 2021.

[4] Yajing Gao, Shixiao Guo, Jiafeng Ren, Zheng Zhao, Ali Ehsan, and Yanan Zheng. An Electric Bus Power Consumption Model and Optimization of Charging Scheduling Concerning Multi-External Factors. *Energies 2018, Vol. 11, Page 2060*, 11(8):2060, 8 2018.

[5] J Wang, I Besselink, and H Nijmeijer. Battery electric vehicle energy consumption prediction for a trip based on route information. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 232(11):1528–1542, 9 2018.

[6] C J J Beckers, I J M Besselink, H Nijmeijer, and J J M Frints. Energy consumption prediction for electric city buses. *13th ITS European Congress*, 6 2019.

[7] Camiel Beckers. Energy Consumption Prediction for Electric City Buses: Using Physics-Based Principles. (PhD Thesis), 6 2022.

[8] Jianhua Guo, Yu Jiang, Yuanbin Yu, and Weilun Liu. A novel energy consumption prediction model with combination of road information and driving style of BEVs. *Sustainable Energy Technologies and Assessments*, 42:100826, 12 2020.

[9] Cedric De Cauwer, Joeri Van Mierlo, and Thierry Coosemans. Energy Consumption Prediction for Electric Vehicles Based on Real-World Data. *Energies*, 8:8573–8593, 2015.

[10] Cedric De Cauwer, Wouter Verbeke, Thierry Coosemans, Saphir Faid, and Joeri Van Mierlo. A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions. *mdpi.com*, 2017.

[11] Teresa Pamuła and Wiesław Pamuła. Estimation of the Energy Consumption of Battery Electric Buses for Public Transport Networks Using Real-World Data and Deep Learning. *Energies 2020, Vol. 13, Page 2340*, 13(9):2340, 5 2020.

[12] Antti Lajunen, Klaus Kivekäs, Francesco Baldi, Jari Vepsäläinen, and Kari Tammi. Different approaches to improve energy consumption of battery electric buses. *2018 IEEE Vehicle Power and Propulsion Conference, VPPC 2018 - Proceedings*, 1 2019.

[13] Hatem Abdelaty, Abdullah Al-Obaidi, Moataz Mohamed, and Hany E.Z. Farag. Machine learning prediction models for battery-electric bus energy consumption in transit. *Transportation Research Part D: Transport and Environment*, 96:102868, 7 2021.

[14] Hatem Abdelaty and Moataz Mohamed. A Prediction Model for Battery Electric Bus Energy Consumption in Transit. *Energies 2021, Vol. 14, Page 2824*, 14(10):2824, 5 2021.

[15] Xiaolei Ma, Ran Miao, Xinkai Wu, and Xianglong Liu. Examining influential factors on the energy consumption of electric and diesel buses: A data-driven analysis of large-scale public transit network in Beijing. *Energy*, 216:119196, 2 2021.

[16] Jari Vepsäläinen, Kevin Otto, Antti Lajunen, and Kari Tammi. Computationally efficient model for energy demand prediction of electric city bus in varying operating conditions. *Energy*, 169:433–443, 2 2019.

[17] Ali Saadon Al-Ogaili, Ali Q. Al-Shetwi, Hussein M.K. Al-Masri, Thanikanti Sudhakar Babu, Yap Hoon, Khaled Alzaareer, and N. V. Phanendra Babu. Review of the Estimation Methods of Energy Consumption for Battery Electric Buses. *Energies 2021, Vol. 14, Page 7578*, 14(22):7578, 11 2021.

[18] Shefang Wang, Chaoru Lu, Chenhui Liu, Yue Zhou, Jun Bi, and Xiaomei Zhao. Understanding the Energy Consumption of Battery Electric Buses in Urban Public Transport Systems. *mdpi.com*, 11 2020.

[19] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. *University of North Carolina at Chapel Hill*, 1995.

[20] Platform Beheer Informatie Standaarden OV Nederland (BISON). Specificatie TMI8: Actuele ritpunctualiteit en voertuiginformatie Koppelvlak 6. Technical report, BISON, 2020.

[21] Dabao Zhang. A Coefficient of Determination for Generalized Linear Models. *American Statistician*, 71(4):310–316, 10 2017.

[22] Weiliang ZENG, Tomio MIWA, and Takayuki MORIKAWA. Exploring Trip Fuel Consumption by Machine Learning from GPS and CAN Bus Data. *Journal of the Eastern Asia Society for Transportation Studies*, 11:906–921, 2015.

[23] Rüdiger Wirth and Jochen Hipp. CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 1, 2000.

A Project management

This appendix includes the project management plan that was established prior to the development of the EVEE system.

Destination Zero

Energy Consumption Prediction for Battery Electric Buses

Project Management Plan

Berend van den Berg

21-12-2021

DATE	CHANGES
21-12-2021	First draft

This document is written according to ISO16326-2019 [1]

Table of Contents

1. Project Overview

The goal of Destination Zero is to reduce the risks and costs of zero-emission (ZE) bus operations and hence accelerate the overall transition from fossil fuel to zero-emission bus operations. It creates an online data hub that provides live electric bus and charge-point data to OEM's, bus operators, charging infrastructure providers and operational planning solutions, with the aim to create operational excellence in zero emission bus operations.

When different actors in the ecosystem are connected and agree to share data for the collective optimization of the system, great efficiency gains can be realized in terms of costs, risks, and time.

1.1. Purpose, scope, and objectives

The current solution is realized with an onboard device that streams bus data real-time to the cloud. From here, raw data or planning assistance tools can be provided to bus operators, thereby aiding the fleet scheduling process. The overall project consists of several work packages:

1. A high-fidelity energy usage prediction model.
2. Integration of live energy forecasting in an adaptive planning system
3. Generation of early warnings for potential operational disruptions regarding buses completing their active or planned assignments

The scope of the PDEng project described in this document is on work package 1 only: extending the onboard device with capabilities for performing real-time energy and range estimation considering the bus's route information. Within this work package, the following tasks are defined for this project:

1. Realizing a first Energy Estimation System V1 (EES-V1) that employs the estimation model previously developed by TUE [2]. The goal is to make a production ready version of the software, that is scalable and flexible enough to be deployed in buses for further validation of the model's accuracy.
2. Verification and validation of V1
3. Design of EES-V2, that is based on a machine learning algorithm, rather than the physical model employed in [2].
4. Verification of a prototype version of V2

1.2. Assumptions and constraints

Throughout this project, the developed software is constrained to be deployed on the Owasys Owa450 IoT platform that has already been selected for this purpose.

1.3. Deliverables

The following deliverables are considered throughout the project:

1. Project Management Plan
2. Software requirements document
3. Test plan for pilot
4. PDEng thesis (confidential)
5. Executive summary
6. Presentation during thesis defense and graduation day

1.4. Schedule summary

DATE	DESCRIPTION
01-11-2021	Project start
19-11-2021	Return day: workshop
26-01-2022	Return day: presentations
22-03-2022	Return day: presentations and supervision
30-05-2022	Return day: presentations and technical reflection
18-07-2022	Return day: presentations and supervision
TBD	Thesis final submission deadline
TBD	Thesis defense
21-10-2022	Graduation ceremony & project end

1.5. Evolution of the plan

The project management plan (PMP) shall be updated after every Project Steering Group (PSG) meeting, if changes were agreed upon. These meetings are approximately once per month.

2. References

[1] ISO/IEC/IEEE: Software and systems engineering, "ISO/IEC/IEEE 16326," no. 2nd edition 2019-12, 2019.

[2] D. Jagga, "Cloud your bus: real-time energy consumption prediction for electric city buses.," TU/e, Eindhoven, 2020.

[3] B. van den Berg, "Project management excel register: risk FMEA, stakeholder analysis, and RASCI matrix," 2021.

[4] B. van den Berg, "Atlassian Confluence Cloud-Your-Bus documentation wiki," 2021. [Online]. Available: https://sycada.atlassian.net/wiki/spaces/GDD/pages/2458943489/Owasys.

[5] B. van den Berg, "JIRA project board," 2021. [Online]. Available: https://sycada.atlassian.net/jira/software/projects/CYB/boards/25.

3. Definitions

3.1. Involved parties

AFFILIATION	NAME	ROLE
Eindhoven University of Technology	Berend van den Berg	PDEng candidate
	Ion Barosan	University supervisor
	Igo Besselink	University supervisor
	Riske Meijer	Program manager
Sycada Mobile Solutions	Kristian Winge	CEO
	Rogier Mulder	Company supervisor

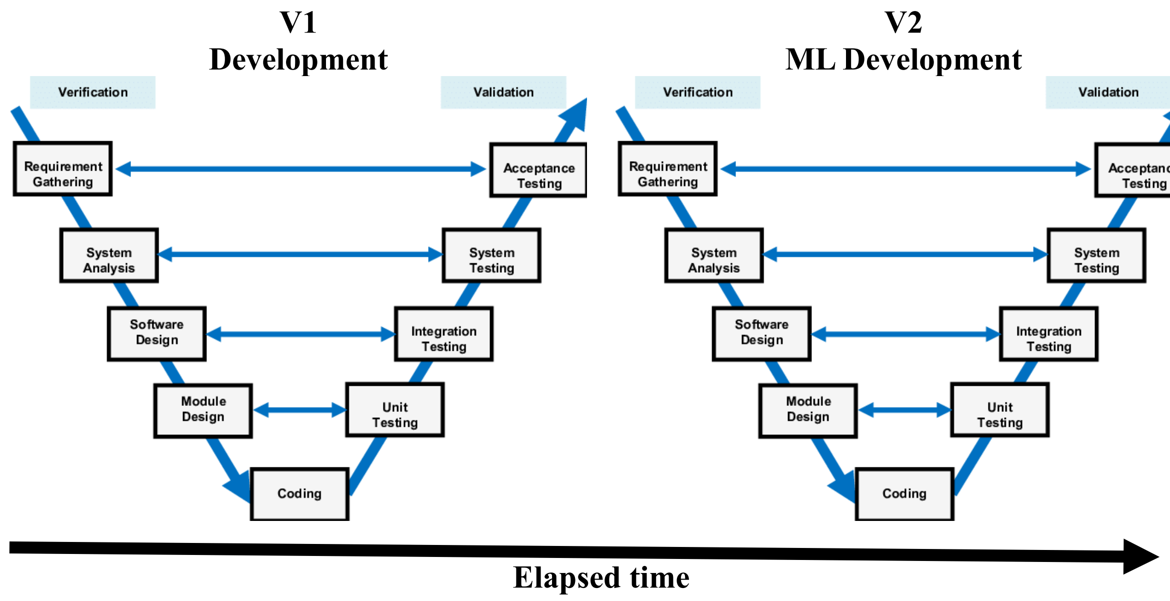
3.2. Terminology

ABBREVIATION	DESCRIPTION
Bitbucket	A git-based source code repository hosting service owned by Atlassian. It offers GIT version control to source code
EES	Energy Estimation System
Jira	Software for issue tracking and project management offered by Atlassian
Confluence	A web-based wiki tool offered by the Atlassian tool suite that facilitates information sharing inside an organization
OEM	Original Equipment Manufacturer
ML	Machine Learning
Version Control System	Method for managing the versions of a software or document
ZE	Zero Emission

4. Project context

4.1.Process model

As described in the project scope, the project consists of the development and verification of two systems. Therefore, a double V-cycle model is employed for the project:



4.2.Infrastructure and enabling systems

The system shall be developed for the Linux-based OWA450 IoT device.

4.3.Methods, tools, and techniques

The following tools and techniques shall be used throughout the project:

- Agile project management with Jira board for issue tracking
- Two-cycle V-model process for development
- Bitbucket for code version control
- CAFCR (viewpoint hopping) and TRIZ methodologies for system architecting
- GoLang shall be mainly used for programming
- Confluence wiki for project documentation

4.4.Product acceptance

The main aspect of final product acceptance is delivering production ready software:

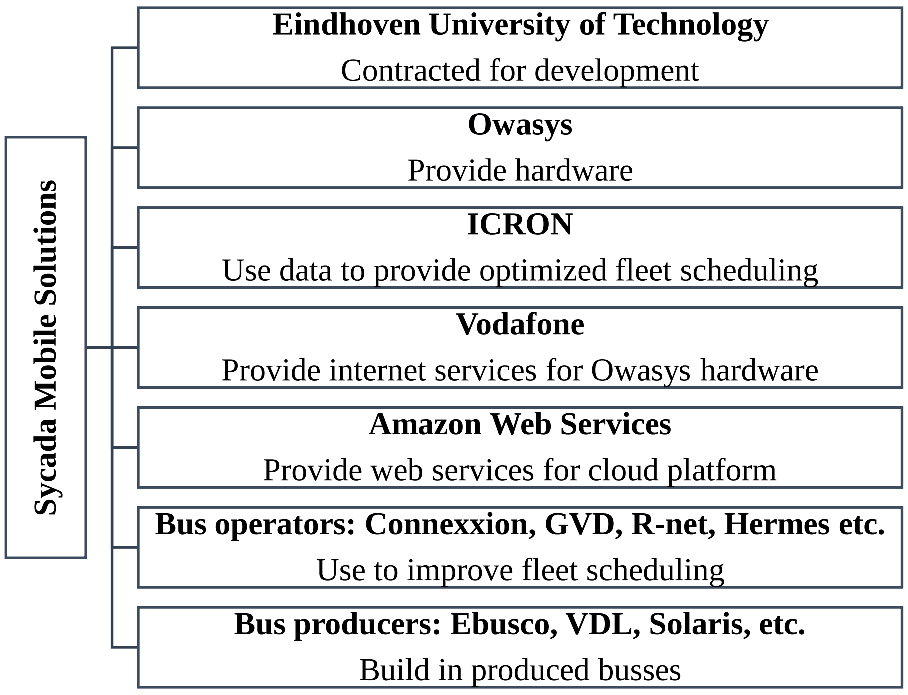
1. Satisfactory estimation accuracy (>95% within 20% of trip duration)
2. The system shall adapt to changes in the environment, to ensure high accuracy over time.
3. Works on any route and bus type
4. Robust and crash-free

To ensure compliance with the stakeholder's requirements, a set of requirements will be established that will continuously evolve in consultation with the project supervisors. Throughout the development of the software, the stakeholder's set of requirements will be the leading factor for the proposed design solutions. A mid-term presentation is planned where progress of product is presented, and compliance will be evaluated.

4.5. Project Organization

This section of the PMP will outline the interfaces, both internal and external, between the development team and third parties as well as define roles and responsibilities for the project.

External interfaces
The external interfaces of the project are presented in the graph below:


Sycada Mobile Solutions
Eindhoven University of Technology
Contracted for development
Owasys
Provide hardware
ICRON
Use data to provide optimized fleet scheduling
Vodafone
Provide internet services for Owasys hardware
Amazon Web Services
Provide web services for cloud platform
Bus operators: Connexxion, GVD, R-net, Hermes etc.
Use to improve fleet scheduling
Bus producers: Ebusco, VDL, Solaris, etc.
Build in produced busses

A detailed analysis of all involved parties, including name, affiliation, concerns, and communication preferences can be found in the stakeholder analysis document [3].

Authorities and responsibilities
Responsibilities shall be determined by breaking down the major work activities into tasks. For each of these tasks, relevant authorities and responsibilities shall be determined in a RASCI matrix, which states the people that are: responsible, accountable, supporting, consulted, and informed throughout this task. The RASCI matrix shall be found in [3].

5. Project Planning

5.1. Project work plans

The following work activities are identified throughout the various phases of the project:

1. **Realize production version of V1**
   - Simulator: realize a simulation setup that runs on a laptop
   - Develop code interface for V1 in GoLang
   - Implement E-rate management service (E-rate initialization & cloud sharing)
   - Review, validate, optimize V1 during operational test/pilot phase
2. **Develop a new version V2 based on Artificial Intelligence**
   - Literature survey
   - Collecting data
   - Development of offline prototype
   - Validation of prototype
   - Optional: Implementation of prototype in real-time estimation
   - Optional: Verification real-time
3. **Comparison of V1 and V2 accuracy**

6. Project assessment and control

6.1. Requirements management

The main platform for exchanging requirements is the company confluence wiki environment [4]. The latest version of the requirements shall be published here to ensure transparency for all parties involved. The requirements shall be linked to the Jira project board [5]. Any changes in the requirements will be communicated as stated in section 8.2.

6.2. Scope change control

When a scope change is proposed by Sycada, the trainee will evaluate the impact of this change on the project within one workday. This impact assessment shall include:

a. The consequences in terms of deliverables
b. Feasibility of timely delivery
c. Updated risk assessment
d. Discussion with TU/e supervisors

These assessments shall be communicated to the project supervisors from both Sycada, and TU/e. and an informed decision shall be agreed upon between all parties involved. Only after confirmation during the PSG meeting, or email communication when this meeting is far away, can the scope change be accepted.

6.3. Schedule control

The backlog of the project is ranked by importance of the features that should be in the delivered software. Backlog items with the highest importance are addressed first. The backlog priority of the backlog items is continuously updated.

6.4. Quality assurance

Quality that should be managed in the projects are as follows:

1. Coding
    a. Execution time
    b. Complexity
    c. Readability
2. Model development
    a. Accuracy
    b. Required hardware resources

To monitor the quality, these points shall be included in the requirements documentation. Throughout the project, this shall be monitored on a regular basis.

6.5. Project closeout plan

During project close out, the following items shall be stored on a cloud research drive, managed by Igo Besselink, with access provided to Sycada:

1. meeting minutes
2. presentations
3. architecture documents
4. final report
5. software product

7. Product delivery

This section describes how the delivery of the deliverables, presented in Section 1.3, will be handed over at project completion. The final version of the software shall be delivered through the Bitbucket repository. A final presentation will be held for the client to showcase the usability and functionality of the software product. All documents that are not yet in the confluence wiki and are relevant for both TU/e and Sycada, will be uploaded into a shared research drive managed by Igo Besselink.

8. Supporting process plans

8.1. Project supervision and work environment

The project shall be mainly executed from the Eindhoven University of Technology. Supervision throughout the project is realized as follows:

1. From Sycada, the main supervisor is Rogier Mulder, who will be responsible for the supervision of the development of the Energy Estimation System (EES).
2. From TU/e, Ion Barosan and Igo Besselink will supervise the project by providing feedback on the direction and content of the project.

8.2. Decision management

This section describes the strategy that is employed when taking decisions throughout the project. The table below describes for each decision type the following things:

1. Who should be involved during the decision?
2. Who needs to be informed about the choice?
3. What actions must be taken before deciding?

Decision about	Involve	Inform	Actions
Project scope	Kristian, Rogier Igo, Ion	Riske	Evaluate consequences regarding planning and deliverables
Project process	Igo, Ion		Check team satisfaction
Requirements	Kristian, Rogier, Ion	Igo	Check conflicts with design Consult Sycada about changes
Software design	Rogier		Evaluate pros and cons Create decision matrix Evaluate consequences for planning

8.3. Risk management

An FMEA analysis shall be performed at the start of the project [3]. In here, internal, and external risks are identified and prioritized according to their Risk Priority Number. During each monthly update meeting with both company and TU/e supervisors, the risks shall be addressed.

8.4. Configuration management

The main platform for code configuration management will be Bitbucket. A cloud-storage folder will be used for other document storage. All documents shall be as follows:

Format:	YYYYMMDD_owner_description_version
Example:	20211201_BB_PMP_V1

8.5. Information management

The project information that is to be managed is:

1. Meeting minutes
2. Project documentation
3. Developed software

8.6. Quality assurance

The quality of the deliverables is monitored throughout the project in the following ways:

1. The quality of the code, being the readability and modularity. Review moments shall be scheduled with the Company supervisor to receive feedback on the work.
2. The overall quality of the program, being its functionality, is evaluated during company stakeholder meetings.

8.7. Measurement

For the verification of the system, the following measures shall be employed:

1. Estimation accuracy: the accuracy of the energy estimation, is determined using the error between the actual energy consumption, and the calculated estimate.
2. Estimation frequency: the frequency at which the energy estimations are calculated

8.8. Reviews and audits

The reviews and audits required to check the progress of the project for visibility, comprehensibility, and acceptability have been outlined in this section. The following reviews and audits are scheduled throughout the project:

1. Monthly meeting with TU/e supervisors
2. Monthly Project Steering Group (PSG) meeting with TU/e and Sycada (2 weeks after 1.)
3. Mid-term evaluation
4. Final evaluation
5. Final presentation

8.9. Verification and validation

The verification and validation procedure contains the following:

1. The verification of the production version of EES-V1 shall be done by establishing a test plan that describes tests for all defined requirements. This comprises of a series of system tests that will be executed according to the test plan, to verify whether the designed software satisfies the customer requirements.
2. The validation of EES-V1 will be done to ensure that the designed system provides the value to the customer as was proposed during project initialization. In this phase, it shall be evaluated whether the provided EES can improve the scheduling of fleets, thereby reducing CAPEX & OPEX.
3. The verification of EES-V2 (based on Machine Learning), shall only include the verification of the prototype software, as described in point (1). Since the main functionality remains the same for V2, validation of the system is covered by point (2).

B Stakeholder Analysis

Within the project there are two main groups of stakeholders: Eindhoven University of Technology (TU/e) (Table B.1, and Sycada Mobile Solutions (Table B.2). A stakeholder analysis was done to identify the preferred level of communication, as well as the main concerns of the stakeholders.

Name	Affiliation	Position	Project role
Ir. Riske Meijer	TU/e	EngD program manager	Manager
Dr. Igo Besselink	TU/e	Associate professor	Supervisor modeling
Dr. Ion Barosan	TU/e	Assistant professor	Supervisor systems design
Yuzhe Ma	TU/e	PhD candidate	Energy modeling support

Table B.1: TU/e stakeholders

Name	Affiliation	Position	Project role
Rogier Mulder	Sycada	CTO	Supervisor company
Kristian Winge	Sycada	CEO	Advisor

Table B.2: Sycada stakeholders

Besides the directly involved stakeholders presented in the tables, additional external stakeholders can be identified.

Since the aim is to offer the developed solution to bus operators to enhance their operation, these should also be considered during the development of the solution. Therefore, the operators interest (e.g. Transdev, GVB, and Connexion) are also considered during requirement elicitation.

To further analyze the influence of stakeholders on the project, a Power Interest Matrix was created in Figure B.1.

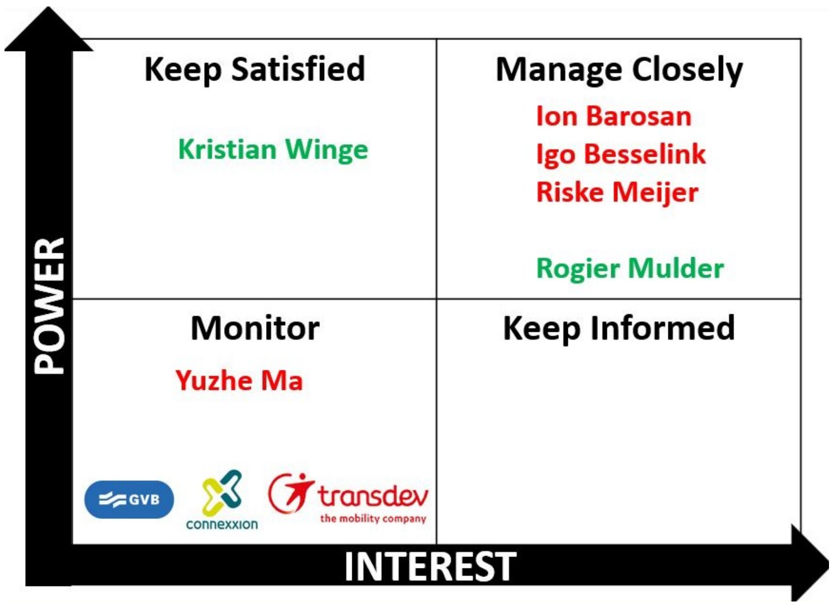


Figure B.1: Stakeholder Power Interest Matrix. Stakeholders from Sycada are marked green, and stakeholders from TU/e in red.

C Requirements tracing

A requirement tracing chart was constructed to ensure that the generated requirements address at least one of the customer needs, This chart is presented in Figure C.1.

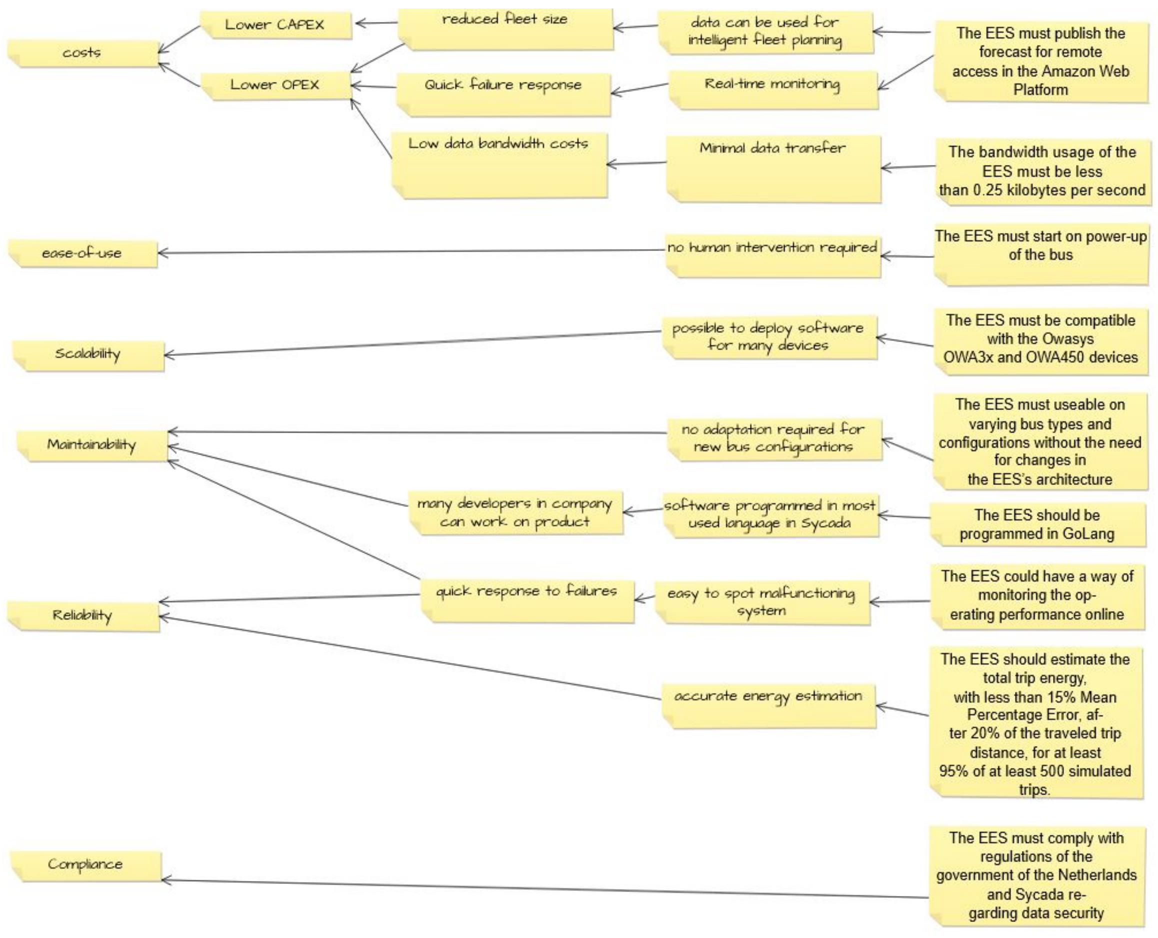

costs
Lower CAPEX
Lower OPEX
reduced fleet size
Quick failure response
Low data bandwidth costs
data can be used for intelligent fleet planning
Real-time monitoring
Minimal data transfer
The EES must publish the forecast for remote access in the Amazon Web Platform
The bandwidth usage of the EES must be less than 0.25 kilobytes per second
ease-of-use
no human intervention required
The EES must start on power-up of the bus
Scalability
possible to deploy software for many devices
The EES must be compatible with the Owasys OWA3x and OWA450 devices
Maintainability
no adaptation required for new bus configurations
The EES must useable on varying bus types and configurations without the need for changes in the EES's architecture
many developers in company can work on product
software programmed in most used language in Sycada
The EES should be programmed in GoLang
Reliability
quick response to failures
easy to spot malfunctioning system
The EES could have a way of monitoring the operating performance online
accurate energy estimation
The EES should estimate the total trip energy, with less than 15% Mean Percentage Error, after 20% of the traveled trip distance, for at least 95% of at least 500 simulated trips.
Compliance
The EES must comply with regulations of the government of the Netherlands and Sycada regarding data security

Figure C.1: Tracing requirements to stakeholder concerns

D Energy consumption parameters literature overview

Parameters influencing energy consumption. An X in a source's column indicates that a correlation is explicitly mentioned in the document.

Type	**Description**	**Symbol**	**[12]**	**[15]**	**[16]**	**[22]**	**[13]**
Vehicle	Mass	m	X	X	X	X	X
Vehicle	Powertrain Efficiency	η	X	X	X	X	X
Vehicle	Roll Coefficient	C_r	X		X		X
Vehicle	Drag Coefficient	C_d	X				
Vehicle	Frontal Area	A	X				
Vehicle	State of Charge	SoC			X		X
Operational	Auxiliary Power	P_{aux}	X		X		X
Operational	Speed/acceleration profile	v		X		X	X
Operational	Regenerative Braking	B_{reg}					X
Operational	Mechanical Braking	B_{mech}					X
Operational	Passenger Load	m_{pas}					X
Topology	Trip Length	D_{trip}	X	X	X	X	X
Topology	Stop Density	S		X		X	X
Topology	Road condition	r_c					
Topology	Road Slope	α		X			X
External	Ambient Temperature	T_a	X		X		
External	Air Density	ρ	X				
External	Headwind speed	v_{wind}	X		X		
External	Gravitation	g	X	X	X	X	X
Derived parameters							
Vehicle	Battery Temperature	T_{bat}					
Vehicle	Battery Capacity	C_{bat}					
Vehicle	Tire Air Pressure	ρ_{tire}					
Operational	Time	ToD		X			
Operational	Weekday	DoW					
Operational	Month	ToY					
Operational	Acceleration Profile	a					
Topology	Maximum Speed	v_{max}					
Topology	Speed bump density	N_{bump}					
External	Air Humidity	h					

E Reference RCI method [1] trip simulation results

Extracting the trips from the two months of logged data provided approximately 2000 trips containing 16 different routes, with a minimum of 50 trips per route. These trips were sorted into training and validation data sets, with per route 30 trips for training, and at least 20 trips for validation. In total, 1334 validation trips were simulated, and the reference trip was updated after each trip.

Performance metric

The performance of the algorithm is defined as the MPE, and is calculated as in Equation E.1, where $E_{estimate}$ is the energy estimate, and E_{actual} is the actual total trip energy consumption.

$$MPE = \frac{100\%}{N} \sum_{n=1}^{N} \frac{E_{estimate,n} - E_{actual,n}}{E_{actual,n}} \tag{E.1}$$

Results

Figure E.1 presents the total energy consumption for all simulated routes. It shows the Absolute Percentage Error of the projected energy estimate, for each route averaged over all its trips. It represents the mean, and std for each route individually, and the error of all routes combined.

The results in Figure E.1 show that the accuracy goal is not reached (see Section 2.3). For identifying the root cause of the large error, the drivetrain estimate errors (see Figure E.2), and auxiliary estimate errors (see Figure E.3) were plotted individually. In this figure it is obvious that the drivetrain estimate error is relatively low compared with the auxiliary error, which has a mean error peaked at 100%.

Moreover, analyzing the entire dataset revealed that on average, the drivetrain component contributes 80% to the total energy consumption, and the auxiliary component the remaining 20%. Hence, the conclusion of the validation is that the auxiliary part of the estimation does not perform well enough for a production-ready version of the software. Improvements should be realized in the auxiliary estimation part.

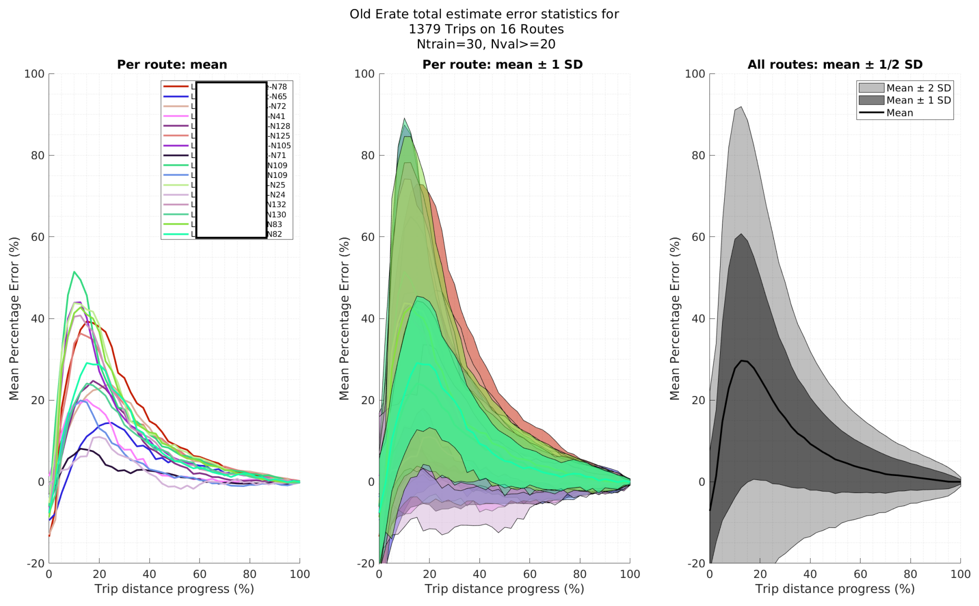


Figure E.1: Initial RCI model: total energy consumption error. Absolute Percentage Error per route. The legend shows the [LineID-StartLocation-StopLocation-NumberOfTrips].

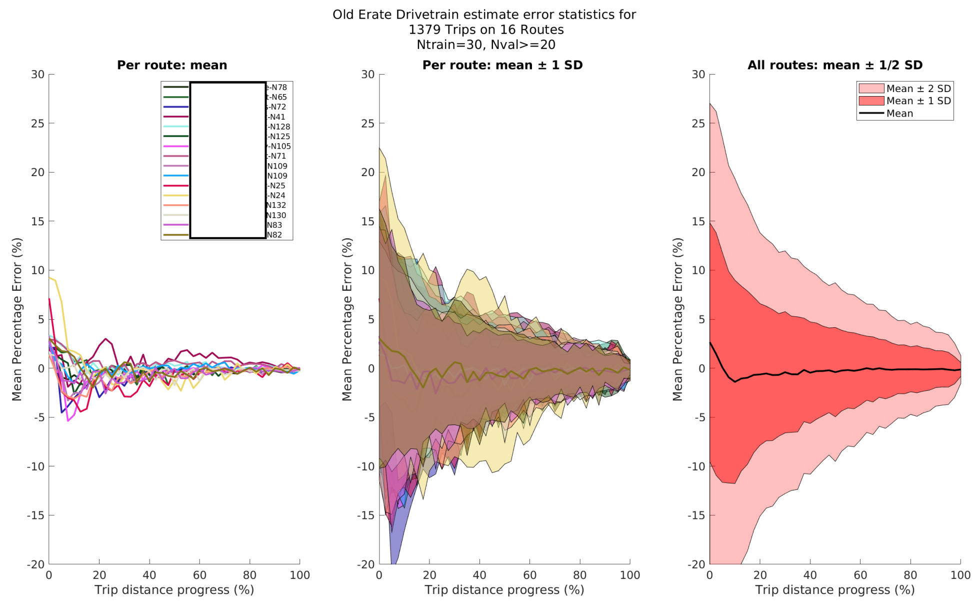


Figure E.2: Initial RCI model: drivetrain energy consumption error. Absolute Percentage Error per route. The legend shows the [LineID-StartLocation-StopLocation-NumberOfTrips].

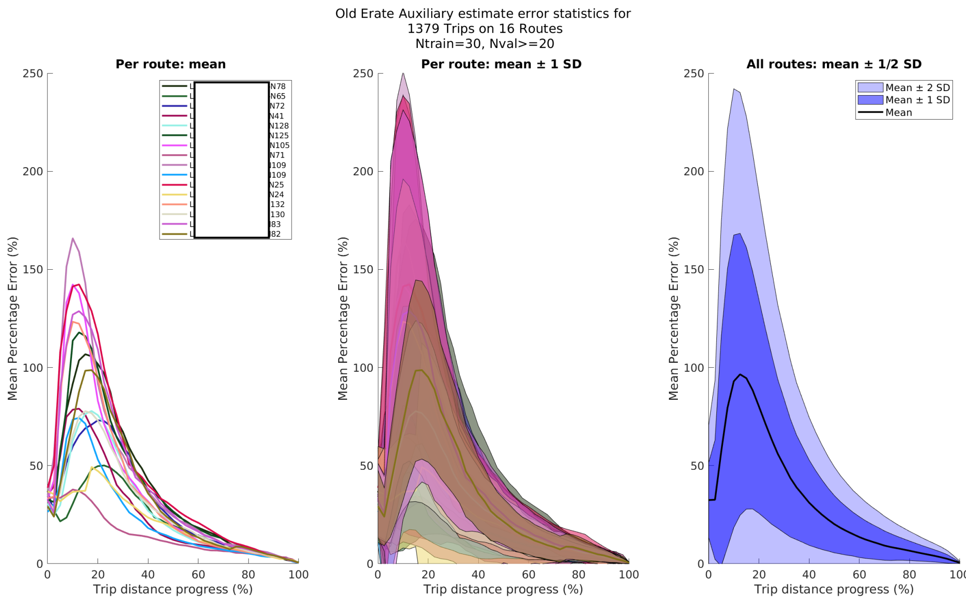


Figure E.3: Initial RCI model: auxiliary energy consumption error. Absolute Percentage Error per route. The legend shows the [LineID-StartLocation-StopLocation-NumberOfTrips].

F RCI simulation algorithms pseudo-code

The algorithm works as follows: first, it loops over all the data and extracts the data for each unique line-trip combination. It then looks up the GPS start and stop coordinates and location names from a reference table. The extracted data is trimmed to start and stop and the right coordinates. If the quality of the data is OK, it is saved in its line-start-stop folder and used during simulations.

Algorithm 1 Trip data filtering algorithm

```
Input:   Input and output directories
         Schedule excel sheet
         Stop location GPS lookup table
Output:  filtered data sorted in line-start-stop folders
for all files do
    $D$ = load(datafile)
    for all unique lines (L) in D do
        $D_{line}$ = filterLine($D$, $L$)
        for all unique trips (T) in $D_{line}$ do
            $D_{trip}$ = filterTrip($D_{line}$, $T$)
            $D_{split,i}$ = splitData($D_{trip}$,$t_{split}$)
            for all i do
                check trip day
                check trip duration ($d \geq t_{min}$)
                $GPS_{start}$, $GPS_{stop}$ = lookupGpsFromRotationTable($table$, $L$, $T$)
                $D_{trim}$ = trimStartStop($D_{split,i}$, $GPS_{start}$, $GPS_{stop}$)
                if all checks OK then
                    saveDataAsLineStartStop($D_{trim}$)
                end if
            end for
        end for
    end for
end for
```

Algorithm 2 Data preparation script

```
      Input:   Minimum samples $N_{tr}$ in training data
               Minimum samples $N_{val}$ in validation data
               input and output directories
      Output:  data sorted in training and validation
               Erate profiles in validation folders
 1: $N_{erates} = 0$
 2: if output directory exists then
 3:     Remove output directory
 4: end if
 5: for all route folders do
 6:     makeDirectory(output/training/routeid)
 7:     makeDirectory(output/validation/routeid)
 8:     for all trip data files ($i$) do
 9:         $D$ = load(datafile)
10:         if $N_{erates} \leq N_{training}$ then
11:             $Erate_i$ = generateErateFromTrip($D$)
12:             copyTripToTraining($D$)
13:             $N_{erates} = N_{erates} + 1$
14:         else
15:             copyTripToValidation($D$)
16:         end if
17:     end for
18: end for
19: $mergedErate$ = mergeTripsIntoErate($Erate_1$ to $Erate_N$)
20: save merged Erate in validation folder
```

Algorithm 3 Automated simulation

```
      Input:   Directory of validation data
      Output:  Energy consumption plots with mean and standard deviation
 1: for all route folders do
 2:     for all validation data do
 3:         load vehicle data
 4:         load main Erate
 5:         run simulink model
 6:         generate Erate with trip data
 7:         if New Erate is okay then
 8:             Update main Erate
 9:         end if
10:         save simulation data in structure
11:     end for
12: end for
13: plot total energy consumption
14: plot inidividual drivetrain and auxiliary energy consumption
```

G CRISP-DM process

The ML model was developed following the CRISP-DM [23], see Figure G.1. This process model consists of the following steps. The process starts with understanding the underlying business problem that the model aims to solve. Next, data is collected and its quality is analyzed. After this, iteratively the data is prepared and a model is established. This is a continuous process during which many features will be created and evaluated in combination with different modeling methods. When the model is completed, it is evaluated by determining whether the underlying business problem is solved sufficiently. Lastly, when approved the model is deployed into the production environment. When in service, the model may be needed to update periodically to include new data.

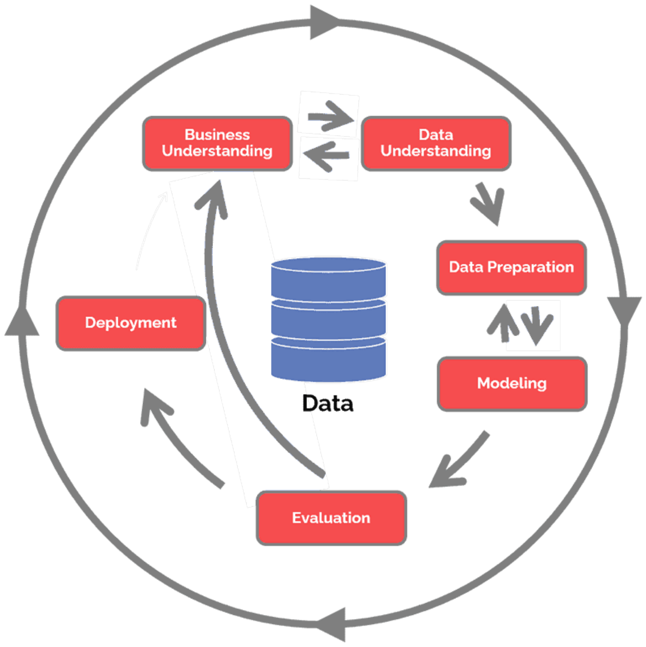


Figure G.1: CRISP-DM: CRoss Industry Standard Process for Data Mining. From 'What is CRISP DM?' by Data Science Process Alliance, 2022 (https://www.datascience-pm.com/crisp-dm-2/)

H Investigating ML estimate volatility

The data accumulator is a system that is present in the current Cloud Your Bus system. It averages the driving data throughout the ride until a driver change event is triggered. The accumulator then resets.

Simulations demonstrated that using the accumulator as input for the machine learning model resulted in extremely volatile energy consumption estimates. Investigation of this issue revealed that this is caused by the convergence of the parameters throughout the trip. When the accumulator just starts averaging the trip data, a lot of volatility is present in the parameter signals. The longer the ride, the less these parameters vary since new driving data has less influence on the averaged signal.

Figure H.1 demonstrates the convergence of these parameters. The examples here are acceleration bands, energy ratios, and speed bands.

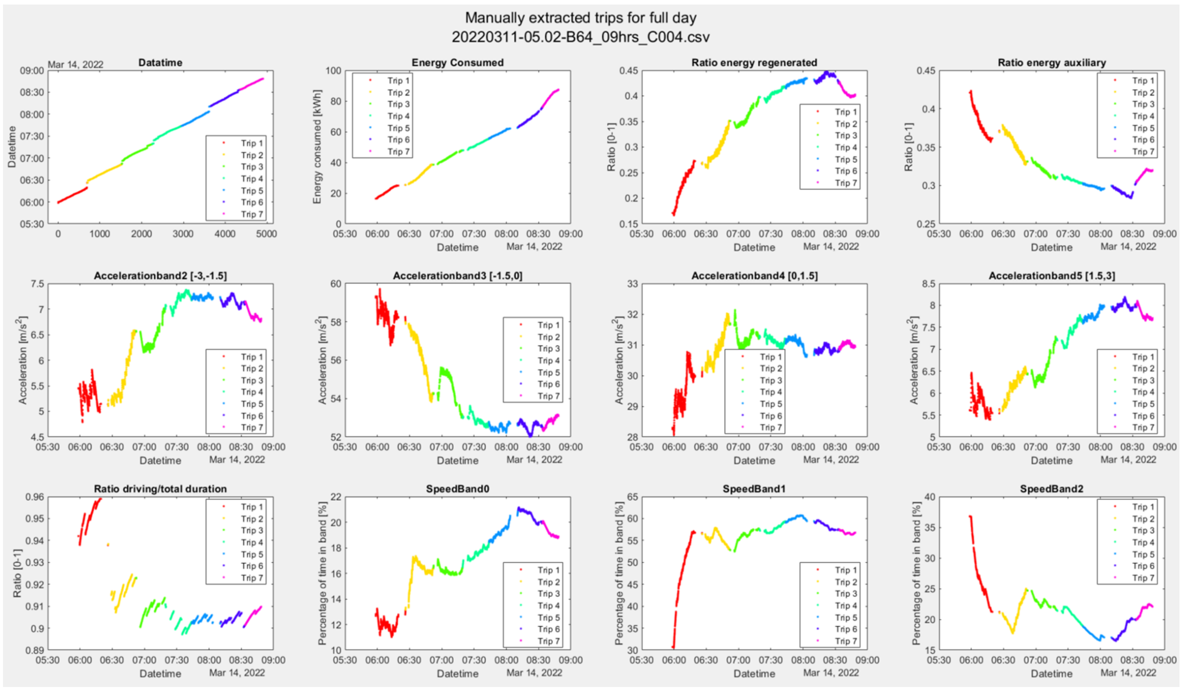


Figure H.1: ML data accumulator during consecutive trips, for energy ratios, speed bands, and acceleration bands. Each color indicates a unique trip.

I Additional information on ML model development

This appendix contains the following additional information on the Machine Learning model development:

1. Description of the estimation concepts that were evaluated
2. Definition of the data points available in the data
3. Analysis of normalized energy consumption per route
4. Discussion of some noticed correlations in the data
5. Presentation of the performed sensitivity analysis

Estimation concepts

The estimation concepts operate with the previously identified variables that have a high correlation with the target variable, which is the trip energy estimate. Several considered concepts concepts are presented in Table I.1, and differ in the following ways. The data complexity indicates what parameters are employed, with the high level being parameters that can be determined pre-ride, and low level includes driving data. The estimation approach either models the combined energy usage, or the split auxiliary and drivetrain components.

Name	**Data complexity**	**Estimation approach**
Concept 1	High level	Auxiliary and drivetrain
Concept 2	High level	Total
Concept 3	Low level	Auxiliary and drivetrain
Concept 4	Low level	Total

Table I.1: Machine Learning modeling concepts overview.

The combined estimator in Equation I.1 employs a ML model that predicts the total energy per kilometer. Here, E_{trip} is the total trip energy, s the distance, E_m the measured energy up to that distance, $\bar{E}_{total/km}$ is the normalized total energy consumption, and D_r the remaining distance of the trip.

$$E_{trip}(s) = E_m(s) + E_{total/km}(s)D_r(s) \tag{I.1}$$

The split estimation in Equation I.2 employs two estimators which estimate drivetrain energy per kilometer ($\bar{E}_{drive/km}$), auxiliary power (P_{aux}). The equation is extended with a term that includes a prediction of the remaining time t_r.

$$E_{trip}(s) = E_m(s) + \bar{E}_{drive/km}(s)D_r(s) + \bar{P}_{aux}(s)t_r(s) \tag{I.2}$$

Both the combined and split estimators are combined with a real-time correction, which combines the normalized predictions with the measured values, by calculating the weighted average of both, see Equation I.3. Here, N_c is the auxiliary or drivetrain normalized estimate. The same correction is employed for both estimates. W is the weight used for calculating the weighted average, $N_{predicted}$ is the estimate, and $N_{measured}$ is the measured value.

$$Nc(s) = (1 - W(s))N_{predicted} + W(s)N_{measured} \tag{I.3}$$

Here, the weight is defined by Equation I.4, and is a measure of confidence that we have in the measured value. This function returns starting with a value of 0 during the first phase of the trip, until it passes the initialization distance p_{init} (we used $p_{init} = 0.15$). From there, it linearly increases up to 1 at 100% trip progress, indicating we fully trust the measurement. The progress of the trip is denoted by $p(s)$.

$$W(s) = max\Big(0, \frac{1}{1 - p_{init}}\Big)(p(s) - p_{init}), \quad p(s) = \frac{s_{current}}{s_{triptotal}} \tag{I.4}$$

Datapoints in ML datasets

Category	**Parameter**	Unit
Energy	Total	kWh
	Auxiliary	kWh
	Drivetrain	kWh
	Regenerated	kWh
Battery	SoC at start	%
	SoC at stop	%
Environment	Ambient temperature	°C
Operations	Distance	m
	Duration	s
	Idle time	s
	Rolling time	s
	Driving time	s
Vehicle	Speed	% time in bin
	Acceleration	% time in bin
	Gaspedal	% time in bin
	Motor rpm	% time in bin
	Engine load	% time in bin
	Battery load	% time in bin
	Artemis Speeds	% time in bin
Driving events count	Hard and excessive acceleration	-
	Hard and excessive braking	-
	Full throttle	-
	Mechanical braking	-
Driving aid display	Green led on time	%
	Yellow led on time	%
	Red led on time	%
GPS	Latitude	°
	Longitude	°
	Heading	°
Bus stop	Start description	-
	Stop description	-
Trip info	Rotation ID	-
	Trip ID	-
	Line ID	-

Table I.2: Data points available in the ML datasets.

Analysis of normalized energy consumption per route

The data understanding phase aids in the selection of an appropriate model, as well as suitable parameters for the input variables. Derived from the project goal, the target variable can be set in two ways. Firstly, a combined energy estimate could be modeled, and secondly a split auxiliary and drivetrain estimate. Therefore, the auxiliary and drivetrain energies were analyzed separately. Figure I.1 compares the average auxiliary power and drivetrain energy normalized per unit of distance), plotted per route in box plots. Here, the auxiliary power plot has a noticeable spread in the data, while the means seem to vary little when comparing the routes. Moreover, the drivetrain plot shows similar data spread, though the means seem to vary slightly more compared to the auxiliary power plot. This could indicate that especially modeling the drivetrain energy could benefit from using data per line instead of aggregated data points.

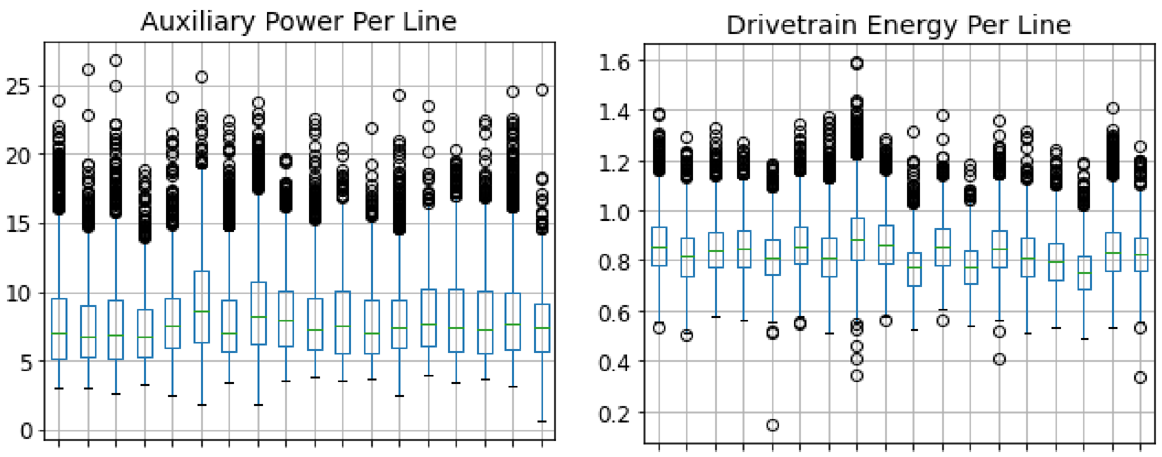
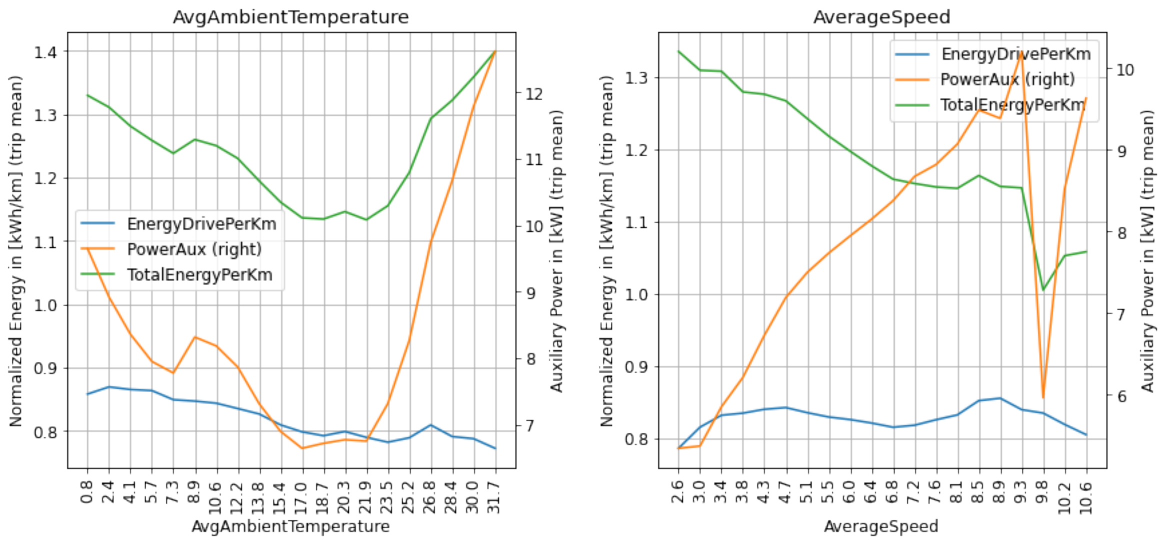


Figure I.1: ML model: Box-plots of normalized auxiliary power and drivetrain energy per route. A route is identified as a combination of a line, with a start and stop location.

Noticeable correlations in trip segments dataset

The first correlation concerns the effect of the ambient average temperature on the normalized auxiliary power, presented in Figure I.2a. From this graph, it seems that the auxiliary power varies significantly with ambient temperature, which may be explainable since the major part of this power is consumed by the heating and air conditions systems. Hence, at low temperatures, the heating is consuming energy, while at high temperatures the air conditioning is activated, which explains the minimum of around 20 degrees.

Moreover, an interesting relation was noticed by plotting average trip speed against auxiliary power consumption, see Figure I.2b. The noticeably higher auxiliary power at speeds may be experienced as surprising since driving is included in the drivetrain energy, which shows relatively less correlation. An explanation for their effects could be that additional energy is provided in the auxiliary portion for cooling the battery pack, and the bus travels at low speeds which makes aerodynamic drag forces negligible.



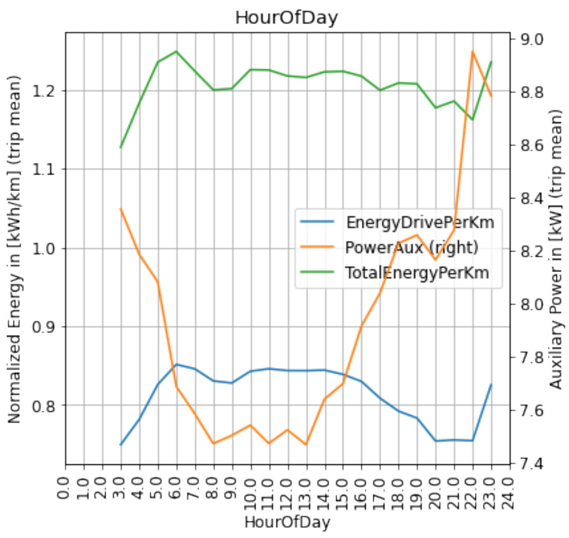
(a) Average Ambient Temperature [°C]

(b) Average Speed [m/s]

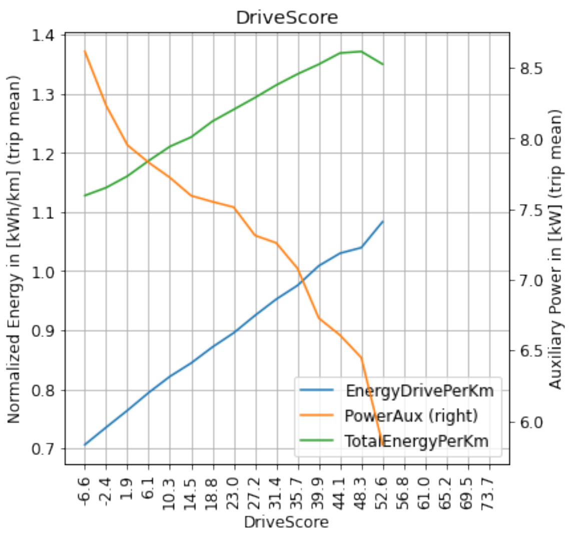
Figure I.2: ML model: correlation of normalized energy consumption with other parameters.

A third correlation can be noticed when plotting energy against the hour of the day, see Figure I.3a. Here, the drivetrain energy appears to peak between 6 am and 16 pm, which could be due to extra traffic on the road, and more passengers on the bus, and thus a higher total mass (Figure I.3a). Additionally, the auxiliary power consumption has a minimum between 8am and 14 pm, and rapidly rises after that.

To conclude, the driving score, which is a measure of driving aggressiveness, shows correlated with the auxiliary power in this dataset. Hence a high score indicates an aggressive driving style with excessive acceleration and braking. The extra energy consumption with a high drive score could be explained by additional battery heat generated due to the aggressive driving style, which needs to be cooled.



(a) Hour of day



(b) Drive score

Figure I.3: ML model: correlation of normalized energy consumption with other parameters.

Sensitivity analysis

Model transparency is an important factor for evaluating a ML model (see Section 5.1). The developed model is considered a black box, and hence the model itself cannot directly explain how it behaves for varying inputs. This is an important factor since this influences the selection of the final parameters used in the model, and helps in reducing the sensitivity of the model to external noise.

To address this issue, a sensitivity analysis was done that establishes insight in the behavior of the developed model, which works as follows. First, the raw training dataset is used for establishing a baseline model. Then, for each of the inputs, a noise of three times the Standard Deviation of that variable is added (-3 STD or +3 STD), while keeping the other variables the same as used for the baseline. Again a model is trained, however now on the modified training data. Lastly, the performance of the new model is compared with the baseline by calculating the MAPE between both predictions. This process is repeated for all variables.

The results of the sensitivity analysis of the final model are presented in Figure I.4. From this analysis, it is clear to see that the estimate is extremely sensitive to changes in the variables representing the ratios of Drivetrain, Auxiliary, and Regenerated Energy compared to the Total Energy.

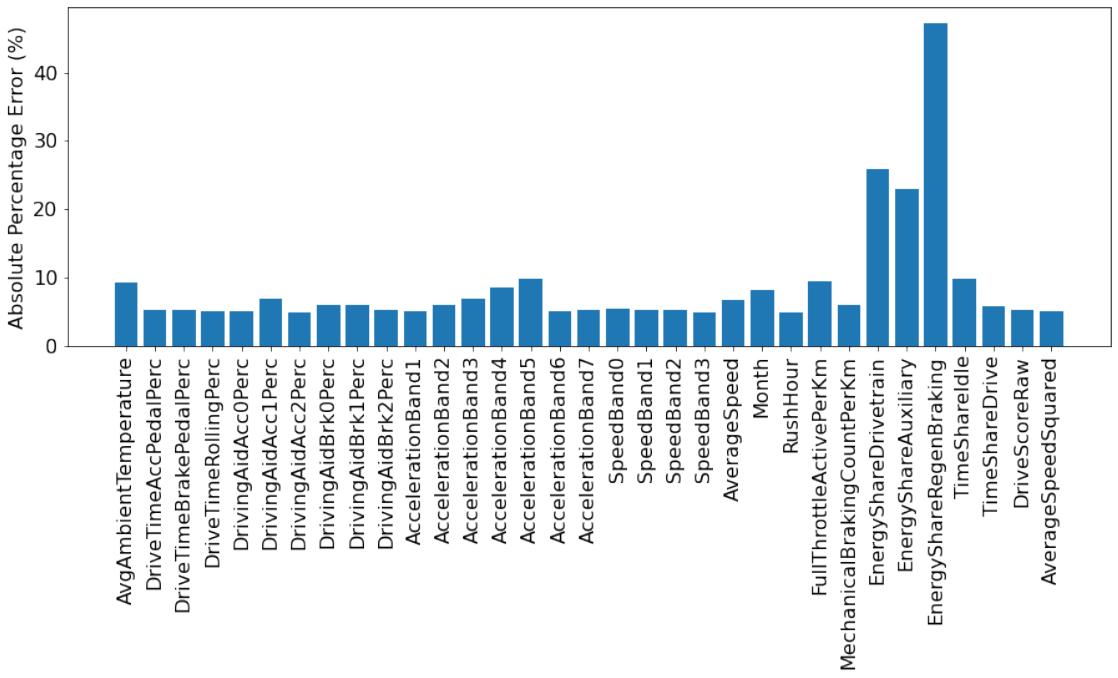


Figure I.4: ML model: sensitivity analysis example for a trained model.

J Additional information on concept selection

Hardware specifications

This section presents a brief overview of several specifications of the Owasys OWA3x[1] and OWA450[2], which are relevant to the concept selection.

	OWA3X	OWA450
Processor	ARM9 400Mhz	ARM Cortex A8 32bit 800MHz
Internal storage	64 MB	1GB
RAM Memory	32 MB	512MB
Internet	GSM (2G)	LTE (4G)
GPS Updaterate	4 Hz	10Hz
GPS Accuracy	2.5 m	2m

Table J.1: Overview of relevant hardware specifications of the OWA3x and OWA450

From this data, it can be seen that the OWA450 has significantly more storage capacity than the OWA3X, which is noticed with the current product because its storage is almost completely full (only 10MB left). Moreover, it has double the computational resources, and many multiples of ram available. It has a more accurate GPS receiver, which is updated at a higher frequency.

The initialization time of the RCI method is constrained by the time required for preparing the reference trip. Due to storage constraints, and the desire to exchange references between buses, the decision was made to store RCI references in the cloud. Downloading the reference characterizes the initialization time of the software. To test this, experiments were done by downloading a 500KB reference on both devices. Results revealed a download time of approximately 7 minutes for the OWA3x, while the OWA450 did so in several seconds.

[1]Owasys OWA3x, accessed 31/08/2022, at https://www.owasys.com/en/products/owa3x
[2]Owasys OWA450, accessed 31/08/2022, at https://www.owasys.com/en/products/owa450

Concept cost calculation

The cost calculations were done using the following assumptions:

- Costs are calculated only for GSM data usage and the required AWS IoT connection.
- Data price is €0.03/MB
- A bus drives approximately 5700 km/month (checked for bus 9501)
- Network overhead data usage is 4%
- Variables are stored with 16 decimal precision (%.16f)
- RCI sizes for the concepts are determined experimentally by creating test messages

Server side estimation

The server-side estimation concept sends all the data points to the cloud, calculates the estimate there, and sends the request back to the client device. For this concept, the costs were calculated by measuring the average amount of data points sent per kilometer. Using this, and the assumption that buses drive between 5000 and 7000 km monthly, the monthly costs of sending the data are calculated for each bus. Table J.2 presents an overview of the estimated monthly costs per bus.

	5000 Km/Month	7000 KM/Month
Maximum	€1.17	€1.63
Practical	€0.64	€0.89

Table J.2: Data transfer costs for server-side estimation

A detailed description of the calculations is as follows. The device requests an estimate each minute while moving. It packs (at most) 60 data points with each request encoded with protocol buffers. Each time the odometer progresses 5 meters, a data point is captured. SAE J1939 defines an odometer update at 1Hz so if the vehicle is driving faster than 22 km/h, it will miss 5-meter-ticks. In a test with the 9501 in EHV, it shows that 54% of the ticks are captured.

Based on a measurement of 1954 datapoints grouped per 60, the average size compressed 2174 (compression ratio of 45.5%). This results in the following on-the-wire size of an estimation request: (2174+40+17)*1.04=2320 bytes. The server-side estimator answers with a response of 30 bytes. On-the-wire this results in (30+40+17)*1.04=90 bytes. This leads to a total of 2320+90=2410 bytes per exchange. Per 100km, there are maximally 20000 and practically 54% * 20000 = 10800 data points. This means between 333 and 180 exchange based on one request per 60 seconds.

The additional costs of keeping the AWS IoT Core connection alive is the same for all options (250+40)*1.04 = 302 *3*24*31 = 0.64MB

Client side estimation

For the client-side estimation concept, the major costs arise from downloading the reference trip from the cloud. Several possibilities were evaluated. First, a raw text file, second a compressed text file (Gzip), and lastly a Protobuf serialized message. The results of the cost calculation are presented in Table J.3.

Transfer concept	reference trip size [KB/km]	data usage [MB/month]	cost [€/month]
Raw rext file	16	190	€5.70
Compressed text file	4.9	57	€1.72
Protobuf message	16.1	191	€5.72

Table J.3: Data transfer costs for client-side estimation

The device downloads a reference trip from storage updates it locally and sends an updated version back to storage. There are four arrays that mainly determine of the reference trip size:

1. RCI
2. Time Remaining
3. GPS latitude and longitude

These arrays have a datapoint every 5 meters. Hence, for a trip of 1km, it consists of 4x(1000/5) = 800 variables. Furthermore, we assume that:

- Data price is €0.03/MB
- A bus drives approximately 5700 km/month (checked for bus 9501)
- Network overhead data usage is 4%
- Variables are stored with 16 decimal precision (%.16f)
- RCI sizes for the concepts are determined experimentally by creating test messages

$$Datausage[MB/month] = \frac{1.04 * [distance/month] * [KB/distance] * N}{1000} \tag{J.1}$$

$$Cost[eur/month] = datausage[MB/month] * pricing[eur/month] \tag{J.2}$$

K Delay testing KV6 schedule messages

Analysis of historical trip data recorded with the embedded device demonstrated situations where the trip schedule change did not match with the actual start or stop coordinate of that trip. To investigate this potential source of system failure, an analysis was done to determine the delay in receiving and processing the KV6 schedule messages.

The message flow of the KV6 schedule is as follows. The trip start or stop message is initiated by a device that is mounted in the bus (not a Sycada device), and sends this message to the BISON servers. The Sycada servers receive this schedule message through a developed interface with the BISON KV6 API. From here, the schedule is sent to the device. See Figure K.

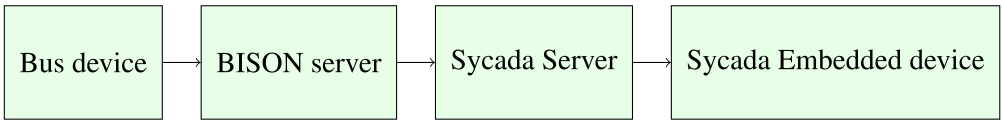


Figure K.1: BISON KV6 delay testing process.

The hypothesis is that the source of the mismatch is the delay between the bus device, and the sycada embedded device. This hypothesis was tested using the timestamp in the KV6 message which indicates the trigger of the event in the bus. With this timestamp, GPS data logged by the embedded device, the coordinates of the event are determined. The results of the tests are presented in Figure K.2.

In the right figure, the orange circles indicate the delay in receiving the KV6 message. This is the time is takes from the bus to reach the Sycada server. The blue circles indicate the timestamp difference with the closest GPS coordinate point that was logged by the embedded device. We only plotted KV6 messages that have a GPS coordinate point with a timestamp that deviates less than 2 seconds.

The left figure shows the GPS coordinates of the start and stop events as black circles. The central station, depot, and several stops (A-F) are marked on the map. This plot shows that the vast majority of the 4.5k messages plotted starts or stops close to the actual locations. However, a minority of messages arrives positions that are far from the trip start or stop.

The conclusion of the tests is that the majority of the messages arives close to the correct locations. However, a small number of messages arrives at an incorrect position. This will happen during operation, and should be dealt with appropriately in the production version of the estimation software.

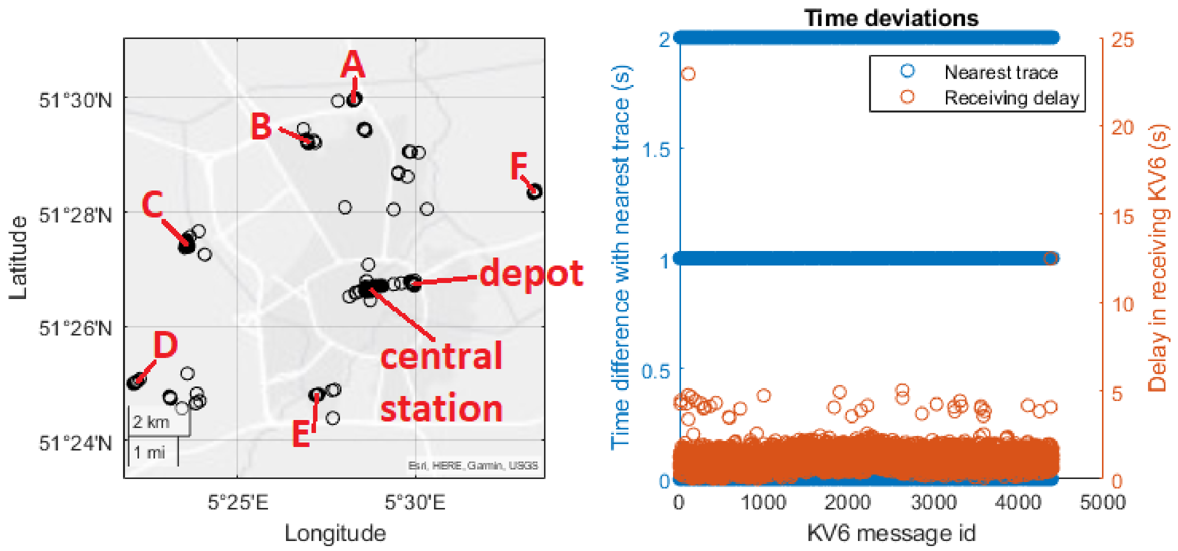


Figure K.2: BISON KV6 start and stop events.

L Additional information on the validation

This appendix contains additional supporting materials for the following requirements:

- Estimation message frequency
- Up-time of estimation
- Data costs

Estimation message frequency

The message frequency was determined by calculating the interval between estimated message timestamps per device. The resulting intervals are presented in Figure L.1. The result is an interval of 60.0 ± 2.5 seconds.

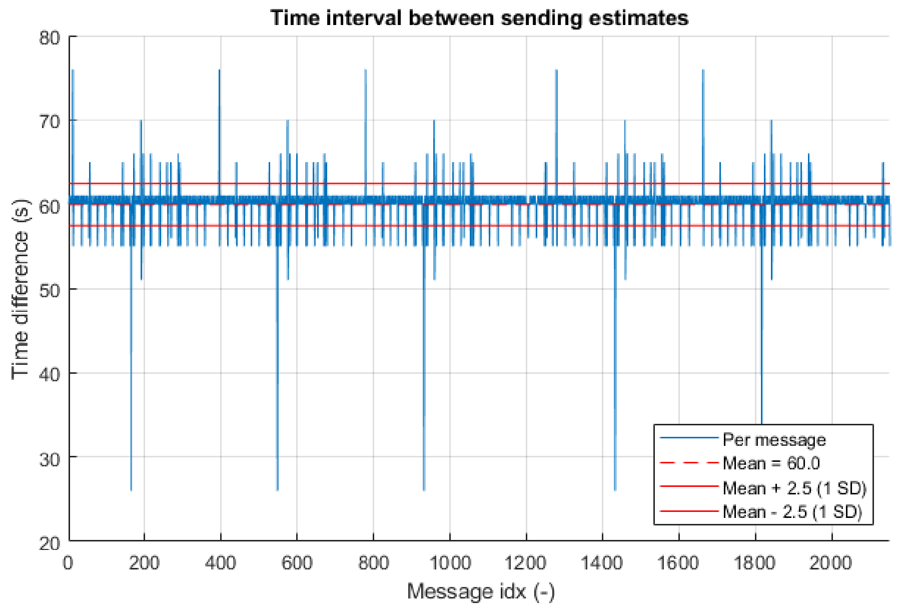


Figure L.1: Estimate message frequency during a pilot test of two weeks.

Estimation up-time

The up-time was evaluated by parsing the log data over a period of two weeks during the pilot test operation. The trips were counted which were successfully initialized with a reference trip, and thus were performing the estimation. The up-time was evaluated based on the daily average, which is presented in Figure L.2. The average up-time was 94.2% during this test.

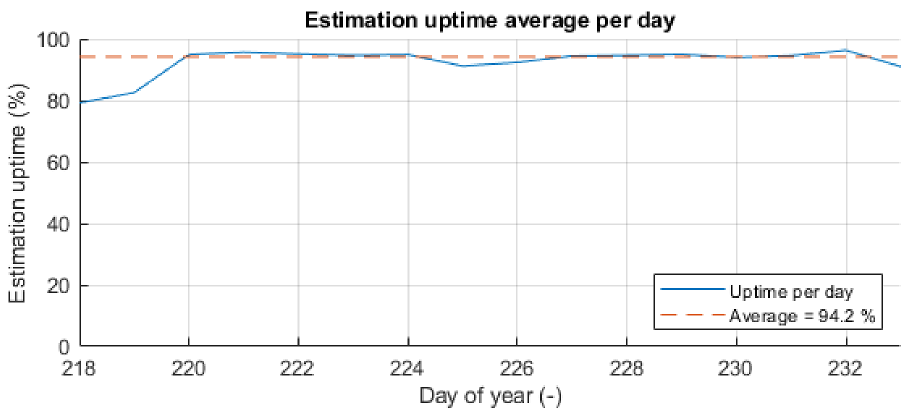


Figure L.2: Estimation up-time during a pilot test of two weeks.

Data costs

The data costs for EVEE were determined with the full implementation, which logged the requests for a total of 16 days for 42 buses. This includes approximately 99% of all request data. The updated communication protocol is defined by the following Google Protobuf messages:

```
enum MessageType {  TraceType = 0;
    EstimateRequestType = 1; EstimateResponseType = 2;}

message RemoteHeader {
    google.protobuf.Timestamp time = 1;
    string originator = 2;}

message EnergyEstimateRequestDatapoint {
    google.protobuf.Timestamp deviceTime = 1;
    float gpsLatitude = 2; float gpsLongitude = 3;
    float vehicleSpeed = 4;  int64 vehicleOdometer = 5;
    float evBatteryEnergy = 6; float evDrivetrainEnergy = 7;}

enum estimationResult {Estimated = 0; CannotEstimate = 1;  Error = 2;}

message EnergyEstimationResponse {
    RemoteHeader header = 1;  string id = 2;
    estimationResult estimationResult = 3;
    string tripID = 4;  string lineID = 5;
    string blockID = 6;  string vehicleID = 7;
    float estimationRemaining = 8;
    float estimationTotal = 9;
    float estimationConfidence = 10;}

message EnergyEstimationRequest {
    RemoteHeader header = 1; string id = 2;
    repeated EnergyEstimateRequestDatapoint datapoints = 3;}
```

The size of these messages is evaluated by averaging 1 million compressed test messages with randomized inputs of similar value to the real messages:

Message	Size	Unit
Request size	2807 + 40 + 17 = 2864	bytes
Reponse size	86 + 40 + 17 = 143	bytes

Table L.1: Request and response message sizes.

Using these messages, and the data obtained from the logged data requests, the following calculations are made to estimate the costs:

```
dataPointCount = 15.153.097;
dayCount=16; daysInMonth=30.4;
busCount=42; datapointsPerRequest=60;
requestsPerBusPerMonth= (datapointCount/datapointsPerRequest)*
                       (daysInMonth/dayCount)*(1/busCount) = 11.439

requestSizeMBytes= 2864/1e6; responseSizeMBytes = 143/1e6;
mbPerMonthPerBus = requestsPerBusPerMonth*
                  (requestSizeMBytes+responseSizeMBytes) = 34.4;

pricePerMb = 0.03; % GSM data bundel
priceTotalPerMonth = mbPerMonth*pricePerMb = €1.03
```

The overall results of these calculations are presented in Table L.2.

Description	Value	Unit
Datapoints/16 days	15.153.097	datapoints/16days
Device count	42	-
Datapoints per device	360.788	datapoints/16days
Requests per device per month	11.439	request/month
Datapoints per device per sec	0.26	datapoints/sec
Data per device per month	34.4	MB/month
Cost per device per month	**€1.03**	€/month

Table L.2: Data cost calculation results.

About the Author

Berend van den Berg received his bachelor's degree in Automotive Engineering, and his master's degree in Mechanical Engineering at the Delft University of Technology, where he specialized in Vehicle Dynamics and Control. During his thesis, he researched a generic motion planning algorithm for autonomous driving in urban and highway driving scenarios. Afterward, he pursued his Engineering Doctorate in Automotive Systems Design at the Eindhoven University of Technology, expanding his theoretical knowledge with the necessary skills to take on challenging and innovative technology projects.