Edge AI System for EV Trip Analytics: On-Device WLTP Class 3B Processing and Battery Temperature Prediction

Lappeenranta–Lahti University of Technology LUT

Master’s Programme in Electric Transportation Systems, Master's Thesis

2025

Nabeel Hasan

Examiner(s): Professor Pedro Nardelli

Dr. Arun Narayanan

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Energy Systems

Electric Transportation Systems

Nabeel Hasan

Edge AI System for EV Trip Analytics: On-Device WLTP Class 3B Processing and Battery Temperature Prediction

Master's thesis

2025

68 pages, 21 figures, 13 tables and 7 appendices

Examiner(s): Professor Pedro Nardelli and Dr. Arun Narayanan

Keywords: Edge Computing, TinyML, WLTP Drive Cycle, Battery Temperature Prediction, Regression Model, Electric Vehicle Monitoring, Energy Efficiency Metrics, Coasting Detection, MQTT Communication, Edge Impulse

The thesis presents a novel edge-native architecture of real-time analysis of driving performance of an electric vehicle (EV) based on a low entry ESP32-S3 microcontroller. The device calculates the important parameters like battery temperature, periods of coasting, battery current efficiency, energy consumption (Wh/km) and regenerative generating energy collection by executing the conventional Worldwide Harmonised Light-duty Test Procedure (WLTP) Class 3B drive cycle. On-device deployment of an Edge Impulse trained TinyML regression model that predicts the battery pack temperature 2 minutes before being depleted is also an important contribution. All the commutated measures are sent over MQTT to a cloud-based Node-RED dashboard to be displayed easily. The framework shows high effectiveness, which has its inference latencies of less than 2 ms and low memory footprint is demonstrated, showing that on-device intelligence is an alternative option to standard cloud-based analytics to enhance EV safety and driving economy.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Pedro Nardelli, for his important assistance and continuous support during this project. Despite his tight schedule, he thoroughly analyzed my initial proposal, offering vital insights and support that set the groundwork for this endeavor. His belief in my work was a continual source of inspiration.

I'm also grateful to Dr. Arun Narayanan for providing organized comments and critical recommendations during our frequent sessions. His guidance was invaluable in refining the study and producing high-quality outcomes, particularly during difficult circumstances.

My deepest gratitude goes out to my brother, Zohaib Hasan, for his steadfast emotional and financial support during my master's program. His faith in me was a source of strength anytime I felt down. I'm also immensely thankful to my parents, Ahmed Hasan and Parveen Akhtar, for their unwavering support in all aspects of my life.

I'd also like to thank my wife, Jasrah Iqbal, for her tolerance and encouragement during this educational journey. My 4-year-old daughter, Dania Hasan's smile provided me with endless energy and delight.

Finally, I want to thank my friends and senior colleagues, Abuzar Akhtar, Wassam Yaseen, and Arham Ali, for their advice, support, and faith in me.

SYMBOLS AND ABBREVIATIONS

Roman characters

m	Vehicle mass	[kg]
C_{RR}	Rolling resistance coefficients	[dimensionless]
A_f	Frontal area	[m^2]
V_{veh}	Vehicle speed	[m/s]
a	Acceleration	[m/s^2]
η_G	Gearbox efficiency	[-]
η_{inv}	Inverter efficiency	[-]
T	Battery pack temperature	[C°]
α	Road slope angle	[deg]
ρ	Air density	[kg/m^3]
δ	Inertia coefficient	[-]

Constants

g	Gravitational acceleration	9.81 m/s^2

Abbreviations

AI	Artificial Intelligence
CAN	Controller Area Network
CSV	Comma-Separated Values
EPA	Environmental Protection Agency

EV	Electric Vehicle
EVS	Explained Variance Score
FTP72/FTP75	Federal Test Procedure driving cycles
HEV	Hybrid Electric Vehicle
IoT	Internet of Things
LTM	Lumped Thermal Model
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
MQTT	Message Queuing Telemetry Transport
NEDC	New European Driving Cycle
SOC	State of Charge
UDDS	Urban Dynamometer Driving Schedule
WLTP	Worldwide Harmonised Light-duty Test Procedure

Table of contents

Abstract

(Acknowledgements)

(Symbols and abbreviations)

Contents

Figures

Tables

1 Introduction

1.1 Background

Electric mobility is now a necessity of the system of modern transportation, and with increased policies on environmental protection and the interest of the society in sustainability matters. The widespread application of EVs indicates a shift away from the conventional ICE technology and faces new tasks regarding performance measurement, efficiency, and safety (Xu et al., 2021a; Liu et al., 2017).

The EV's dynamics are very sensitive to the interactions of multiple variables. Battery temperature, specific energy consumption or Wh/km rates, and the capacity of the regenerative braking system immediately affect the operating range of the vehicle or battery lifespan (Boretti, 2016; Wang et al., 2022). These metrics are also strongly affected by external conditions and driver behaviour. Accelerating quickly, braking hard and speeding up may contribute to the rise in energy consumption, as well as accelerating battery degradation.

Studies have indicated that the application of eco-driving strategies could lead to a substantial reduction in energy consumption (Xu et al., 2021b), with driving behavior considered a critical factor that impacts the efficiency of the vehicle (Braun and Rid, 2018). This underscores the importance of systems that ensure drivers get real-time meaningful feedback that will achieve smoother driving.

The present EV dashboards are, however, limited. Most EV dashboards offer simple displays of the form of the remaining battery percentages. More sophisticated analyses are typically performed on the cloud. Performing analyses on the cloud is not optimal when considering real-time EV analysis because the cloud-based analysis requires connectivity with the network; it introduces the problem of transmission delay; there are connectivity security issues; and there is the issue of continuously sending the data to the cloud. Several researchers state that cloud-based pipelines lack the timing requirements; the reliability

requirements; and the security needs required on the decision-making system (Narayanan et al., 2020; Andriulo et al., 2024).

1.2 The Rise of Edge Computing and TinyML

Edge computing is an alternative feasible protocol to cloud-based processing. In edge computing devices, data are analysed on or around the device of origin and not transferred to remote cloud servers. This design saves on latency, removes network reliance, enhances privacy, and boosts reliability, which becomes especially crucial with EV applications in which data needs analysis at every second (Narayanan et al., 2020).

This is a trend that is increasing at a high rate particularly in Europe. The Edge AI Strategy that has been created by the European Commission indicates that most millions of sensors, vehicles, machines, and energy systems will be independent intelligent nodes instead of relying on cloud computing. Edge platforms are already being adopted by automotive firms, battery manufacturers, robotics firms, and power-system operators to minimise cloud load, enhance responsiveness, and privacy. This action indicates the greater objectives of the EU in terms of independence, sustainability, and safe data processing.

The next promotion of this is Tiny Machine Learning (TinyML). This is achieved by TinyML which enables training machine-learning models on very small microcontrollers, devices with only kilobytes of memory, and with a very small amount of power usage. Due to the efficiency of TinyML models without the use of the cloud, they are implemented in such areas as smart manufacturing, wearables, health monitoring, energy systems, predictive maintenance, and advanced robotics. Industry projections state that the billions of microcontroller-based devices shipped annually will become intelligent edge devices making TinyML one of the biggest spheres of embedded AI (Khan, K. 2025).

Tiny machine Learning (TinyML) builds on this concept offering the possibility to execute machine-learning models on small and low-powered micro controllers. The models used in Tiny Machine Learning must fit within limited amounts of memory while still being useful and can thus be used for real-time analytics on small hardware like the ESP32-S3 (Soy, 2023).

Edge Computing together with TinyML, one can perform advanced energy and thermal analysis within the vehicle without utilizing a cloud system. As part of this thesis, the above technologies are employed to create an on-device system that analyzes driving data, calculates the performance metrics per second, and estimates the battery temperature in real time on the ESP32-S3 chip.

1.3 Problem Statement

The interplay of electrical and thermal issues in operation is a significant influence on the effectiveness of an electric vehicle. There are 5 measurable metrics, namely, battery temperature (C), energy usage (Wh/km), regenerative braking percentage, coasting percentage and battery current (A), that influence the driving range, battery health and cycle life, and long-term reliability. To achieve precise predictions and energy management in EVs, one should understand the way these metrics interact with each other within a short time.

Numerous research examines the energy and thermal characteristics of EVs, and most of them are based on offline modelling or cloud-based calculation, requiring advanced technologies and able to tolerate delays in the transfer of data. Current schemes rarely look at the possibility of implementing the same analytics directly on embedded or edge devices and with limited computational and memory capabilities. The study also does not have uniform information on the per-second data processing that is relevant in registering how people drive and the influence of the same to the temperature. As a result, a limited perception of the trade-offs between the computational feasibility, precision, and the latency at the time of real-time assessment near the data source exists.

This specific issue proposes that the process represented in Figure 1 could potentially map the relationship between factors. This could potentially form a direct causal chain; the issue of the vehicle's Dynamics (acceleration and braking) determines the Battery Current (A), which consequently creates an Energy Consumption (Wh/km) rate and Heat (Temperature) effects. The viability of this process could potentially be confirmed by the direct relationship of the current changes being directly reflected upon energy consumption and energy recovered as shown on our visualization plots.

This is what the core question in the study is: how the low-power advantage system can actualise these interactions under constraint in an unambiguous and consistent manner. The research questions to filling the gap include:

RQ1: To what extent do the 5 core parameters demonstrate their ability to be demonstrated and manipulated on an edge platform using limited resources at 1 Hz (one Nep per second) without exceeding the limits of memory or timing constraints?

RQ2: What is the accuracy-latency trade-offs of making short-horizon predictions of battery-temperature behaviour on-device, and how these predictions are compared to an offline baseline of battery-temperature behaviour using the same data?

RQ3: To what extent do per-second metrics (Wh/km, regeneration share, coasting share, and battery current) capture the effect of the vehicle-level effects (range, battery health, and durability) of WLTP Class 3B profile?

The approach can maintain the focus on the problem and the gap and minimal questions that will inform the remainder of the thesis.

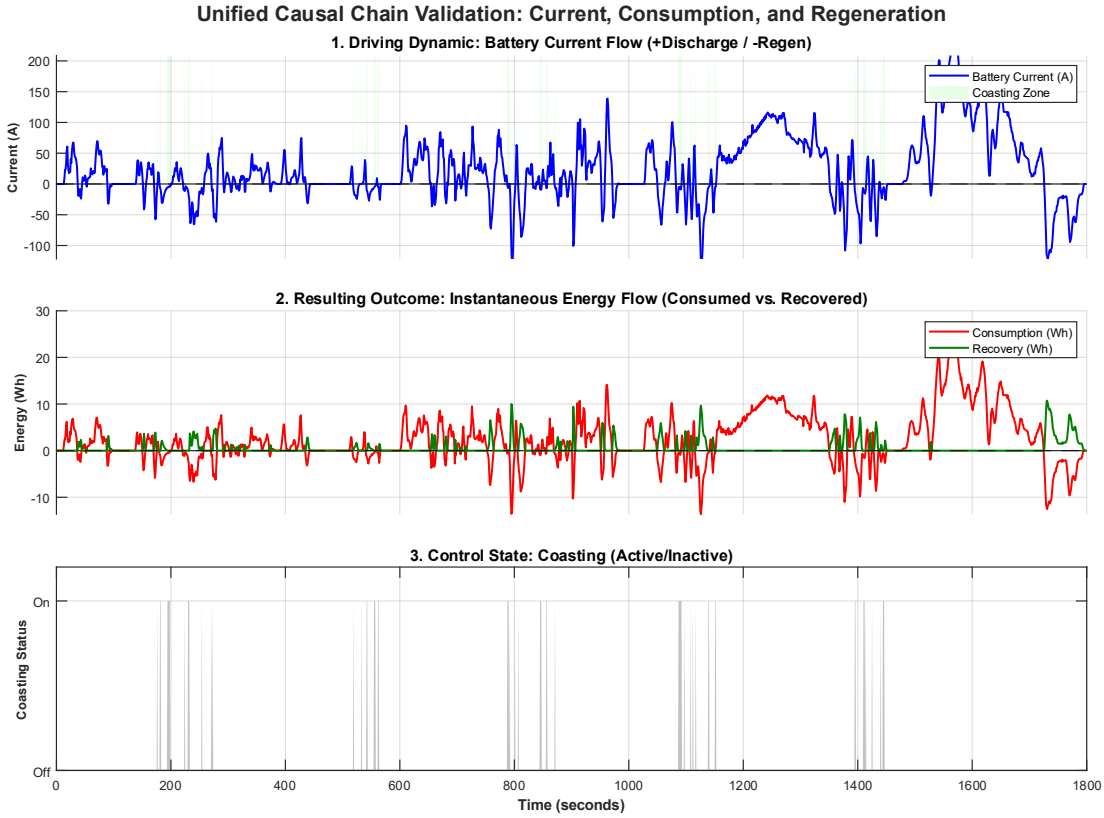

Figure 1: Unified Causal Chain Validation: Current, Consumption, Coasting and Regeneration

1.4 Research Objectives

The major goal of this thesis is to close the highlighted research gap by designing, building, and validating a low-cost, on-device Edge AI system for real-time EV performance measurement. To accomplish this core goal, the following 5 particular goals are established:

1. To provide a sensor-agnostic system capable of extracting all relevant data from a single, standardized 1 Hz speed profile (WLTP Class 3B).
2. To create a complete suite of physics-based models in C++ firmware that is optimized for use on the resource-constrained ESP32-S3 microcontroller.
3. To train and install a lightweight TinyML regression model (TensorFlow Lite) on the ESP32-S3, which can reliably estimate battery temperature two minutes in advance.
4. To ensure the accuracy of all on-device firmware computations, cross-check the findings using an independent offline MATLAB analysis.
5. To create an end-to-end data pipeline that sends real-time analytics from the device to a cloud-based Node-RED dashboard using MQTT for visualization.

1.5 Contribution of This Thesis

This thesis main contribution is the effective design, implementation, and validation of an entire edge-native system on a single ESP32-S3 microcontroller. This work demonstrates that it is possible to run a whole suite of complicated, physics-based analytics and a predictive AI model directly on a low-cost, resource-constrained device, thereby solving the latency and privacy problems of cloud-based solutions.

As evidence of this contribution's innovation, a research article based on the methods and results of this thesis has been approved for publication at the SAI Computing Conference (London, July 2026).

1.6 Thesis Structure

The rest of the thesis is organized as follows:

Chapter 2: Research review examines the scholarly research on key performance measures, including physics-based models for battery current, energy, regeneration, and thermal dynamics.

Chapter 3: Methodology describes the fundamental approach of this work. It details the WLTP dataset, the physics-based techniques employed, the TinyML model's architecture, and the hardware/software configuration of the ESP32-S3 and Node-RED dashboards.

Chapter 4: Results and Analysis summarize and analyses the findings. It compares the on-device calculations to MATLAB, evaluates the TinyML model's predictive performance, and displays the final dashboard and system efficiency data.

Chapter 5: Conclusion summarizes the thesis important results and contributions, evaluates its limits, and suggests prospective areas for future research.

2 Literature Review

Research on the use of electric vehicles has increased in recent years. The purpose of this research is to enhance the efficiency of energy, provide thermal safety, and create better control systems. The literature available is centred around regenerative braking, battery current, coasting, battery temperature and energy consumption.

In this chapter, there is a discussion of important achievements of different fields. This aim is to provide a critical review of the current methods of performance measurement which include coasting detection, battery current and energy calculation. The value of these methodologies and its limitations by evaluating the advantages and limitations will help us identify a critical research requirement which is the aim of our proposed edge-based initiative.

2.1 Acceleration and Coasting Detection

Familiarity with the dynamics of driving, in regard to acceleration and coasting, are important to find out the efficiency of an EV. Power is needed in acceleration. Coasting is the process where the vehicle moves based on its own momentum under with little power needed is a good opportunity to save energy. The fact that these two periods can be accurately determined is, therefore, important in evaluating and forming an effective style of driving.

Acceleration can be computed easily, based upon time-series data of velocity and the underlying physics premise $a = \frac{\Delta v}{\Delta t}$ is applied when calculating the data of velocity. Vehicle sensors, GPS or the Controller Area Network (CAN) bus can receive this information.

The description of coasting is however harder to detect. The great contribution made by Sun et al. (2019) is to develop a control algorithm that identifies coasting. They rely on the data of high-resolution internal vehicle provided by the CAN bus like pedal position and motor torque. Although this type of system is correct, the primary flaw of this system is that it uses proprietary vehicle data. Such CAN signals are often lacking to the researcher or developer,

especially in the context of using public datasets (e.g. WLTP) or in assembling low-cost and non-proprietary embedded systems.

2.2 Battery Current Estimation

Correct estimation of battery current is instrumental in estimating the performance of EV because it impacts power consumption, regenerative braking abilities, and power demand. The current in a battery of an EV is bi-polar: when the number is positive, it denotes an outflow of energy on a battery to energize the motor in traction mode, whereas when the number is negative, it denotes an input of energy on a battery in regenerative braking. The current is not directly under the control of the drivers, but the driving habits of the drivers such as fast acceleration, smooth, anticipatory braking have a massive indirect effect of the current flow, and hence the total energy.

Battery current measurement can be done with the aid of specialized equipment like shunt sensors or via networking with the onboard Controller Area Network (CAN) of the vehicle. Nonetheless, such approaches tend to be practically impossible when used in studies based on open datasets or when implementing a solution that is economically cheap and does not require intrusion on hardware. This limitation requires relying on the reliable methods of estimation.

A common and widely used approach is to find an approximation to battery current using physics-based models based on vehicle dynamics. Miri et al. (2020) designed an underlying model in this field that consists of computing the instantaneous traction power (P_{tra}) required to surmount resistive forces on the vehicle, rolling resistance (R_r), aerodynamic drag (R_a), road grade force (R_θ), and inertial force (R_i). It can be estimated, by modelling these processes, that the estimate of traction power can be obtained based on familiar vehicle parameters (e.g. mass, frontal area) and the current driving conditions (speed, acceleration). Battery current (Ibatt) is obtained by inverting the power value with considering the efficiency of the drive train elements such as the inverter and gearbox among others.

In Miri's work, the traction power P_{tra} is calculated using the equation:

$$P_{tra} = \frac{F_{road} \cdot V_{veh}}{\eta_G} \qquad (1)$$

Where:

F_{road} is the sum of all resistive forces acting on the vehicle:

$$F_{road} = R_\theta + R_R + R_A + R_I \quad (2)$$

Grade force due to road slope:

$$R_\theta = M \,.\, g \,.\, \sin\alpha \quad (3)$$

Rolling resistance is calculated using the equation:

$$R_R = C_{RR} \,.\, M \,.\, g \,.\, \cos\alpha \quad (4)$$

Aerodynamic drag can be estimated by using:

$$R_A = \frac{1}{2} \,.\, \rho \,.\, A_f \,.\, C_d \,.\, V_{veh}^2 \quad (5)$$

Due to acceleration, the inertial force is:

$$R_I = \delta \,.\, M \,.\, a \quad (6)$$

From the above equations, each component of F_{road} is explained in Table I.

Table I: Description of Physical Parameters Used in Traction Power

Symbol	Term	Description
M	Vehicle mass	Mass of the EV (kg)
g	Gravity	Acceleration due to gravity (9.81 m/s²)
α	Road slope angle	Road gradient in radians or degrees
C_{RR}	Rolling resistance coefficient	Constant depending on tyre-road interface
ρ	Air density	Typically, 1.225 kg/m³
A_f	Frontal area	Vehicle's front-facing area (m²)
C_d	Drag coefficient	Aerodynamic drag factor
V_{veh}	Vehicle speed	Instantaneous speed of the vehicle (m/s)
a	Acceleration	Derived from velocity data
δ	Inertia coefficient	Accounts for rotating components, usually > 1
η_G	Gearbox efficiency	Gear system efficiency (0.9–0.97 range)

Once P_{tra} is known, battery current is calculated as:

$$I_{batt} = \frac{P_{tra}}{\eta_{inv} \,.\, V_{pack}} \quad (7)$$

In most empirical studies, the inverter efficiency η_{inv} is (usually estimated at 0.9 to 0.98) and battery pack voltage (V_{pack} usually 400 or 800 V) are used to simulate battery behavior across sample driving cycles. The analytical method to be used in this paper will involve the indirect modelling method which will be used to estimate the second-by-second current using CSV driving data supported by WLTP Class 3B which will not require the use of actual hardware measurements. The most notable benefit is that it makes it possible to model the performance of battery without having to have actual sensors or environment.

2.3 Energy Consumption Estimation

One of the goals of a performance analysis is to estimate the energy consumption of an EV (in Wh/km). The researchers must balance precision and pragmatics including data accessibility and computational expense. This is a very important balance with real-time, on device applications. There are numerous ways that are described in the literature.

Method 1: Laboratory Experiments. Some research employs complicated laboratory sets. Zhang et al. (2023) used special laboratory equipment (a chassis dynamometer) to determine energy will be saved by way of the regenerative braking.

Method 2: Statistical models. In other studies, there are complex statistical frameworks. Indicatively, Chen et al. (2017) required the data provided by 45 individual sensors to determine the trends in the energy utilization.

These methods are reasonably focused, yet they cannot fit on a low-end, scaled back advantage project. They require either expensive lab equipment or a substantial number of sensors, which is not supported by a simple microcontroller.

Modelling in physics gives a much more realistic technology of embedded systems. This approach is fair in balance between accuracy and simplicity of calculations. One such interesting model is the one by Miri et al. (2020). They worked out the quantity of force needed to run the vehicle against physical aspects (air drag and road friction). They found that this model could give strikingly good results when it was tested on regular driving cycles and the inaccuracy that occurred was only 2% to 6% compared to real-world tests.

Miri et al.'s study is significant since it demonstrates that a physics-based model may operate effectively. It is computationally efficient (quick) and sensor-independent (requires few

sensors). This makes it an excellent candidate for IoT and edge computing projects with minimal device resources. As a result, this thesis employs their physics-based approach to quantify energy use in real time, with just the WLTP Class 3B speed profile as input.

2.4 Regenerative Braking Efficiency in Electric and Hybrid Vehicles

The efficiency of regenerative braking, often simply referred to as the percentage of the kinetic energy recovered during braking processes to the available kinetic energy at the vehicle wheels is a key parameter in contemporary electric vehicles (EVs) and hybrid electric vehicles (HEVs). In simple words, it is a measure of how much of the mechanical work that a system uses in slowing itself down can be regained. In its simplest form:

$$\eta_{regen} = \frac{E_{rec}}{E_k} \times 100\% \qquad (8)$$

Where $E_k = \frac{1}{2}mV^2$at the onset of braking, and $E_{rec} = \int P_{regen}(t)dt$. Regenerative braking is an important element in EVs. It grabs the vehicle's kinetic energy as it slows down and turns it back into electrical energy to charge the batteries. This is significant because it enhances driving range, conserves energy, and lowers wear on the mechanical brakes. Researchers utilise a few popular approaches to quantify efficiency:

Method 1: Laboratory Experiments. The most precise way is to use laboratory equipment. Zhang et al. (2023) employed a chassis dynamometer and high-resolution CAN bus data. They discovered that improved control systems may save up to 41% of energy. However, this approach necessitates the use of costly laboratory equipment and specialised CAN bus signals.

Method 2: On-Road Testing. Other research, such as Silva et al. (2019), make use of real-world sensors (such as wheel torque sensors). They discovered that efficiency was highest in city traffic (20%) but lowest on highways (2%). This technology is feasible, but it needs specialised, costly sensors.

Method 3: Computer simulation. Some studies, such as Wang et al. (2021), employ computer simulations to evaluate control techniques. This is adaptable, but simulations are not always completely realistic and may not correspond to real-world hardware.

All these technologies have limits for our project since they require specialised facilities, costly sensors, or private data. This demonstrates a glaring research gap: there is no easy, sensor-agnostic approach for calculating regeneration efficiency in real time using a standard 1 Hz speed file. This thesis presents a physics-based model to close that gap.

Wang et al. (2021) conducted a simulation-based study of a series-parallel hybrid electric vehicle (HEV) and compared two regen-control strategies in a co-simulation (within the AMESim and MATLAB/Simulink platforms). Both the approaches are tested in Worldwide Light-Duty Cycle (WLTC) and the New European Drive Cycle (NEDC) and the energy-recovery performance is measured as

$$\eta_{regen} = \frac{\int P_{regen}(t)\, d(t)}{\int P_{wheel}(t)\, d(t)} \times 100 \qquad (9)$$

Silva et al. (2019) conducted a rigorous on-road test to determine how much energy might be recovered under various driving situations. They drove through city, suburban, and highway traffic in a unique vehicle equipped with wheel-torque sensors and GPS. They discovered that energy recovery was effective in stop-and-go city traffic (about 20%) but decreased to nearly nothing on the wide highway (about 2%). Their findings indicate that real-world testing provides a significantly more realistic image than models. However, the disadvantage of this technology is that it is quite costly and requires specialized sensors.

Other studies conduct meticulous lab testing (known as dyno tests) on production vehicles. Kim & Park (2020) examined two Nissan Leaf models (2012 and 2013). They tested both autos in a lab at various temperatures. They discovered that simple software (firmware) adjustments to the newer model's battery-control system enhanced regen efficiency. The efficiency improved from 70-80% in the previous model to 72-87% in the new one. This demonstrates that even little software modifications may result in significant energy savings.

After examining these findings, an obvious research void emerges. No study has presented a straightforward, sensor-independent (sensor-free) approach for estimating regen efficiency per second. The literature lacks a physics-based solution that can operate in real time utilising simply a normal 1 Hz speed file (such as a WLTP Clas 3B profile) and does not require CAN bus data or expensive lab equipment. Filling this gap is critical for future Edge-AI applications that require drivers to get real-time input. This thesis develops and tests such a technique. Our solution is based on fundamental physics equations and can be confirmed using existing test data.

2.5 Battery Temperature in Electric Vehicles

Battery temperature is a major parameter of electric vehicle efficiency, life and safety. Below the optimum temperature range, lithium-ion cells have an increase in internal resistance, thus decreasing available power and slowing charge acceptance. On the other hand, high operating temperatures increase the rate of unwanted side reactions such as the expansion of the solid-electrolyte interphase (SEI) and the formation of lithium dendrites, which irreversibly decrease capacity and increase cell impedance. Thermal runaway is a possible risk beyond a certain temperature and active cooling systems are required to keep cell temperature within the recommended operating range of 15-35°C.

Lumped Thermal Model (LTM) (Miri et al. (2020) is considered as an electrical analogy of the pathways of heat-flow in a cell. To use the model, two key parameters are determined:

Thermal capacitance C_{th}: The ability of the cell to hold heat, i.e. the energy (in joules) needed to raise the cell temperature by 1 kelvin. C_{th} may be estimated roughly as the product of the mass of the cell m and its specific heat capacity c_p

$$C_{th} = m \times c_p \tag{10}$$

For a typical 50 Ah pouch cell of mass ~0.5 kg and $c_p \approx 900$ J/(kg·K), get $C_{th} \approx 450$ J/K.C

Thermal resistance R_{th}: It is a measure of how easy it is to conduct heat across the interface between the cell surface and the surrounding (i.e. ambient air or a cooling jacket). Such resistance can be isolated experimentally by keeping Q constant and measuring the steady-state temperature increase ΔT.

$$R_{th} = \frac{\Delta T}{Q} \tag{11}$$

In air-cooled packs, typical values fall in the range 0.5–2 K/W, depending on fan or coolant flow.

Once C_{th} and R_{th} values are set, the differential equation is ready to capture two competing processes

$$C_{th} \frac{dT}{dt} = Q_g - \frac{T - T_{amb}}{R_{th}} \tag{12}$$

Heat Generation (Q_g): The heat generated in a battery per unit time Q_g may be expressed as $Q_g = I\,[V_{oc} - V]I + I(t)_R^2$ where $I(t)$ current is being drawn or released. The first term on the right is the irreversible entropy heating caused by loss of overpotential and the second is the purely resistive, Joule-type heating. Combined, these two elements give an accurate reading of the watts of heat that is deposited in the interior of the cell.

Heat Dissipation $\left(\frac{T - T_{amb}}{R_{th}}\right)$: The rate of heat loss of a cell to the surroundings is proportional to the difference between the temperature of the cell and the surrounding. The greater R_{th} the slower this happens; high resistance equals poor cooling so the cell will stay hot longer.

The transient thermal trajectory can be monitored by using discrete incremental $\frac{dT}{dt}$ in small time steps, e.g. 1 s. In the early stages of a heavy-current discharge, the Q_g exhibits a strong peak, and temperature will therefore increase very quickly since the dissipation term is insignificantly small at this stage. As temperature rises, the dissipation term rises until it is balanced by the new equilibrium at which the $Q_g$input is balanced by the dissipative loss, and a new steady state is achieved.

The simple parameter model that is being studied- labelled LTM shows a strong resemblance to the real-world heating and cooling dynamics, in that it can reproduce the relevant time constant ($\tau = C_{th} \times R_{th}$) and the steady-state offset. It is therefore implemented embedded through simple current, charge related voltage and ambient temperature measurements followed by executing a lightweight algorithm in firmware. Mean prediction error has been empirically demonstrated to be about 0.13 K, proving that LTM provides pseudo real-time thermal estimates accurate enough to be used in battery management and edge-AI.

2.6 Embedded Edge AI and TinyML for Automotive Systems

Recent research indicates that shifting analytics from the cloud to the device can minimize latency and dependency on networks, but it imposes tight constraints on memory, timing, and energy. Meuser et al. (2024) explore Edge-AI system topologies and identify the major constraints: tight latency budgets, tiny memory footprints, and device power restrictions. They claim that most research uses a broad systems perspective and that domain-specific, on-device validations remain restricted, particularly beyond vision and voice activities. This

reinforces the necessity to investigate electric-vehicle analytics directly on restricted hardware rather than only in the cloud.

TinyML survey results reach similar conclusions on the model side. Heydari (2025) provides typical ranges for microcontroller deployment, such as model size, on-device inference time, and the accuracy-latency-energy trade-off. The study covers a wide range of application areas; however it provides limited information on automotive driving cycle data or per-second KPI computation for temperature and energy variables. In summary, TinyML is achievable on MCUs, although evidence for EV monitoring at 1 Hz is limited. The reliability of on-device forecasts is also gaining popularity. Zhang et al. (2024) demonstrate that small deep ensembles can enhance uncertainty calibration on edge accelerators, while incurring computation and memory expense. Their findings highlight an important gap for EV use: while calibrated forecasts are achievable, the cost of non-vision telemetry such as battery temperature increase or Wh/km at per-second cadence is seldom assessed. A side-by-side view is shown in Table II, contrasting Meuser et al. (2024), Heydari (2025), and Zhang et al. (2024).

Table II. Comparative snapshot (methods vs. needs vs. limits)

Study	**Focus & method**	**Platform constraints reported**	**Key limitation vs. this thesis**
Meuser et al., 2024	Edge-AI architectures; systems view	Latency, memory energy discussed conceptually	Lacks EV 1 Hz KPI validation on MCU-class hardware.
Heydari, 2025	TinyML on-device inference; survey	Model size, MCU inference time; trade-offs highlighted	Minimal coverage of EV per-second analytics or thermal KPIs.
Zhang et al. 2024	Uncertainty on edge accelerators	Extra compute/memory for calibrated ensembles	No microcontroller-level EV benchmarking at 1 Hz.

2.7 Synthesis and identified gap

Across these various sources, we see three shortfalls for EV analytics on embedded hardware: (i) demonstrations of per-second KPI bookkeeping for the 5 core metrics (battery temperature, Wh/km, regenerative share, coasting share, battery current); (ii) short-horizon

thermal prediction executed on-device with precision observed against an offline baseline; and (iii) explicit budgets for RAM, flash, and 1 s deadlines. These are the queries that this thesis addresses in a standardized WLTP Class 3B environment.

3 Methodology

This chapter describes our sensor-agnostic methodology. This approach is unique in that it only employs a single data source: the public WLTP Class 3B speed profile, sampled at 1 Hz. This eliminates the requirement for specialised CAN bus signalling or a laboratory environment. Using this single speed trace, we use physics-based equations to determine 5 important metrics: coasting time, battery current, energy consumption, regeneration efficiency, and battery temperature. All these measurements are computed every second (1 Hz). They are then merged to get one trip efficiency score. This score is transferred to a cloud-based dashboard, which provides the driver with a comprehensive overview of their driving performance. The dashboard also gives additional information, such as feedback on each measure and short-term battery temperature projections, which will be covered in the next sections.

3.1 WLTP Class 3B Speed Profile Acquisition and Preprocessing

The Worldwide Harmonised Light Vehicles Test Procedure (WLTP) is the latest global standard for testing passenger vehicles. It is far more realistic than the previous standard, known as the NEDC, because it more closely resembles real-world driving. Because WLTP is the new standard, practically every major EV manufacturer, including Tesla, Volkswagen, and Mercedes-Benz, uses it. They utilise this test to report their vehicle's driving range and energy consumption (in kWh/100 km). This enables buyers to simply and properly evaluate the efficiency of various electric vehicles. We selected the WLTP Class 3B profile for this thesis because it is a reliable, repeatable, and industry-standard test.

For this thesis, we imagined that we didn't have an actual vehicle to test. As a result, we adopted a offline-replay strategy. We used the WLTP Class 3B dataset, a basic CSV file. The system read this file one line at a time, one per second (1 Hz), like a live sensor stream. We applied physics-based formulae to determine all the relevant parameters, including traction power, battery current, energy utilisation, coasting, and temperature. These measurements were subsequently transmitted to the ESP32-S3 microcontroller. The ESP32

handles all the final edge calculations immediately on the chip. Finally, the findings were delivered to a cloud dashboard in JSON format for display.

The WLTP Class 3B data is plotted in Figure 2 having 4 distinct segments that add up to a total of 1800 s:

- Low Urban (589 s): Frequent stops and accelerations are in included, average speed ~18.9 km/h, a city-based environment.
- Medium Suburban (433 s): Moderate-speed cruising with occasional slowdowns, average speed ~39.4 km/h.
- High Rural Road (455 s): Steady higher-speed travel on moderate roads, average speed ~56.5 km/h, with few stops.
- Extra-Highway (323 s): Sustained highway speeds up to 131 km/h, average ~91.7 km/h, representing freeway conditions.

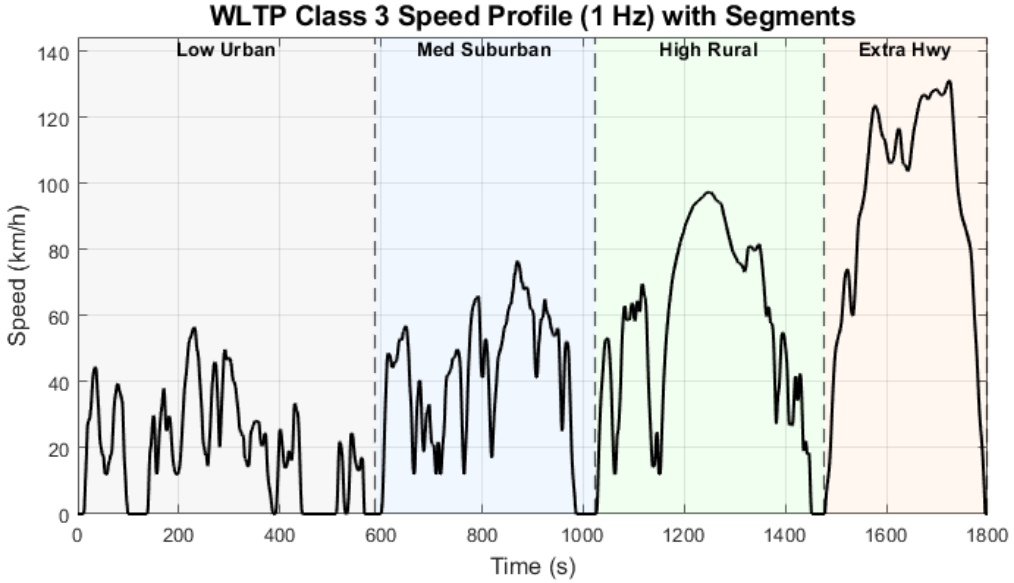

Figure 2: WLTP Class 3B speed profile (1 Hz) with segments

The next sections describe the steps of the procedure necessary to extract each of the parameters out of the original speed trace, as well as the data-preprocessing phase that must be performed before edge-based computation.

3.1.1 Acceleration and Distance Extraction

Acceleration $a(t)$ in m/s^2 is calculated directly by the discrete 1-Hz speed samples $V(t)$ altered from km/h into m/s^2. To calculate the acceleration at 1 Hz, the finite-difference formula is used:

$$a(t) = \frac{V(t) - V(t-1)}{\Delta t} \qquad (13)$$

Breaking down the formula

- $V(t)$ in WLTP Class 3B is the speed in km/h at second, which is converted into m/s^2 by dividing 3.6.
- The numerator states the difference of speed change over one second.
- Dividing by Δt = 1 s gives the acceleration in m/s^2.

Similarly, the distance was calculated as a simple arithmetic operation: the 1 Hz speed (in km/h) is divided by 3600 to be converted in kilometres per second. Thus, each second $\Delta d(t)$ is the distance covered in that time, obtained by repeatedly dividing the sampled speed in km/h by the constant 3600.

$$\Delta d(t) = \frac{V(t)}{3600} \ [km] \qquad (14)$$

Because the sampling interval is Δt is 1 s, the exact distance travelled in that second. These increments were added to provide the cumulative distance at time t:

$$D(t) = \sum_{\tau=1}^{t} \Delta d(\tau) \qquad (15)$$

This is an entirely WLTP Class 3B speed trace-based method generates per-second distance profile of the full 1800 s drive cycle. The total distance covered in WLTP Class 3B dataset is 23.27km with the average speed of 46.5 km/h as shown in Figure 3.

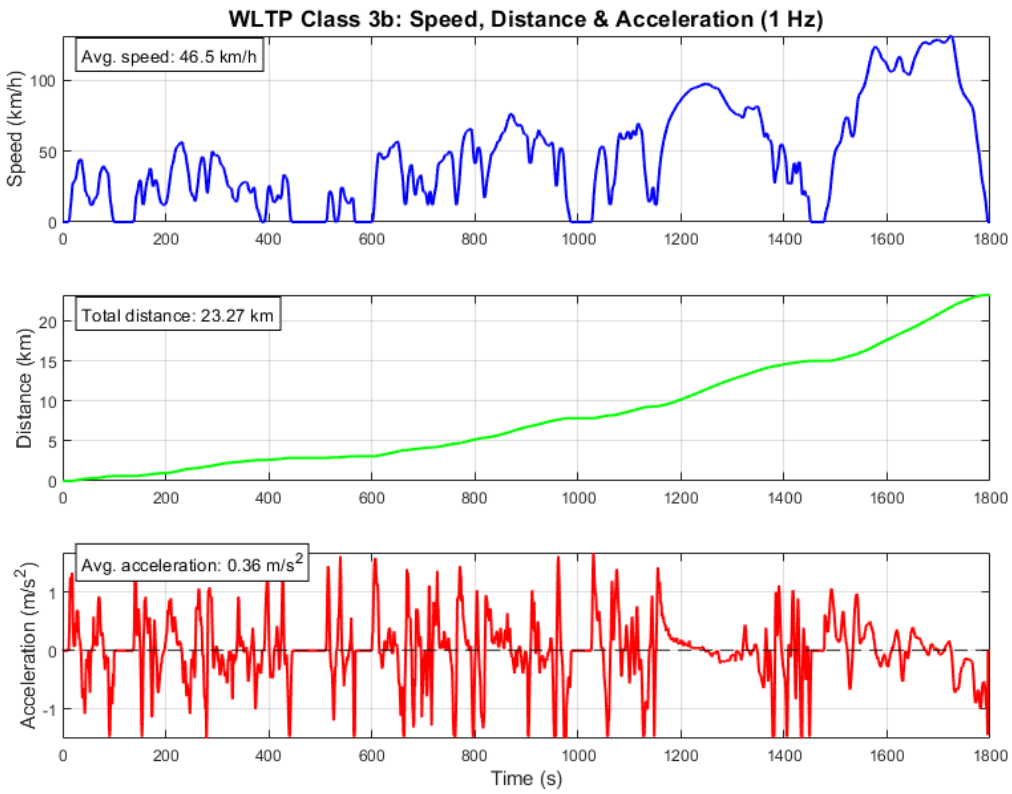

Figure 3: WLTP Class 3B 1 Hz profiles with average speed, total distance, and average acceleration

3.1.2 Importance of Coasting Detection in EV Efficiency

Coasting occurs when a vehicle travels using its own momentum, without the motor accelerating or the brakes being applied. It is an important strategy to increase an EV's driving range. In this mode, the electric motor is not powered, and the vehicle just rolls. This requires little to no battery power.

There are two major advantages to tracking coasting. First, a vehicle's control system may automatically save energy by coasting. Second, we can provide real-time feedback to drivers, encouraging them to drive more smoothly and effectively.

As we highlighted in the literature study, researchers such as Sun et al. (2019) have discovered methods to identify coasting. Their solution, however, needed the analysis of specific inputs from within the vehicle, such as pedal locations, motor torque, and wheel sensors.

This strategy is based on proprietary (private) sensor data and specialised laboratory equipment. These solutions are inappropriate for our project because we didn't have access to the pricey tools or confidential data streams.

In new methodology, to detect coasting segments from 1 Hz WLTP Class 3B driving cycle, a Boolean flag is defined:

$$C(t) = \begin{cases} 1, & if\ |a(t)| \leq \varepsilon_a\ , v(t) > 0, I_{batt}(t) \leq 0, \\ 0, & otherwise, \end{cases} \quad (16)$$

Let's discuss the terms used in this formula.

- $a(t)$ is the acceleration (ms^{-2})
- $v(t)$ is the speed (km/h)
- $I_{batt}(t)$ presents the battery current (A)
- ε_a is just a small acceleration threshold ($0.1ms^{-2}$)

Coasting was determined only when the acceleration $a(t)$ measured is within a range of 0.1 m/s^2; effects of tiny bumps were not considered as coasting. The times in which vehicle speed $v(t)$ is 0 km/h were removed to ensure that only real rolling motions are captured and not idling. The battery current $I_{batt}(t)$ must be zero or negative, so that coasting was distinguished from throttle applications of insignificant magnitude and the drag of slight retardation. These three logical test procedures at 1 Hz, high-fidelity coasting segments were estimated.

3.1.3 Battery Current Estimation from WLTP Class 3B

As described in Chapter 2, Miri et al. (2020) have previously confirmed this physics-based strategy. They utilised the same equations (1–7) to compute traction power and battery current for a 2014 BMW i3 on the NEDC and EPA driving cycles. They tested their findings in a lab using CAN bus data and a chassis dyno, with errors of only 6%. This thesis employs the same well-proven technique. We applied it to the WLTP Class 3B profile. When reading 1 Hz samples, the traction power P_{trac} and $I_{batt}(t)$ were calculated using the same coefficients mentioned in the Miri et al. (2020) with some appropriate changes and shown in the Table III as well.

Table III. Vehicle and Model Parameters for Battery Current Estimation

Parameter	Symbol	Value	Units	Description
Vehicle mass	M	1600	kg	Curb weight of BMWi3 is 1395kg, while with passengers, assuming 1600 kg
Gravity acceleration	g	9.81	m/s^2	Standard gravitational constant
Air density	ρ	1.2	Kg/m^3	Ambient air density
Frontal area	A_f	2.38	m^2	Vehicle front cross sectional area
Drag coefficient	C_d	0.30	-	Aerodynamic drag factor
Inertia factor	δ	1.15	-	Accounts for rotating mass effects
Rolling resistance coeff.	C_{RR}	0.012	-	Tyre road rolling resistance
Gearbox efficiency	η_G	0.96	-	Gear efficiency
Inverter efficiency	η_{inv}	0.92	-	DC-AC conversion efficiency
Battery pack voltage	V_{pack}	355	V	Nominal pack voltage

For computing the road gradient $\alpha(t)$, using sinusoidal function at 1 Hz were used.

$$\alpha(t) = \alpha_{max} \sin\left(\frac{2\pi t}{T}\right) \qquad (17)$$

Where α_{max} is the peak grade with 5% value and T is 300 s period slope between the rising and the falling phase.

We plotted the battery current in MATLAB to gain a good picture of the EV's energy balance. Observing the bidirectional current (two-way current) over time allowed us to clearly identify the discharge phases (positive spikes) and regenerative braking phases (negative spikes). The magnitude of these spikes indicated which elements of the WLTP Class 3B profile are the most energy intensive.

Top Plot (Current vs. Time): This plot depicts the instantaneous current across the whole 30-minute voyage. Stop-and-start city driving is noticeable as abrupt changes from positive (acceleration) to negative (braking). In contrast, the highway phase of the driving cycle exhibits smoother and more consistent positive currents.

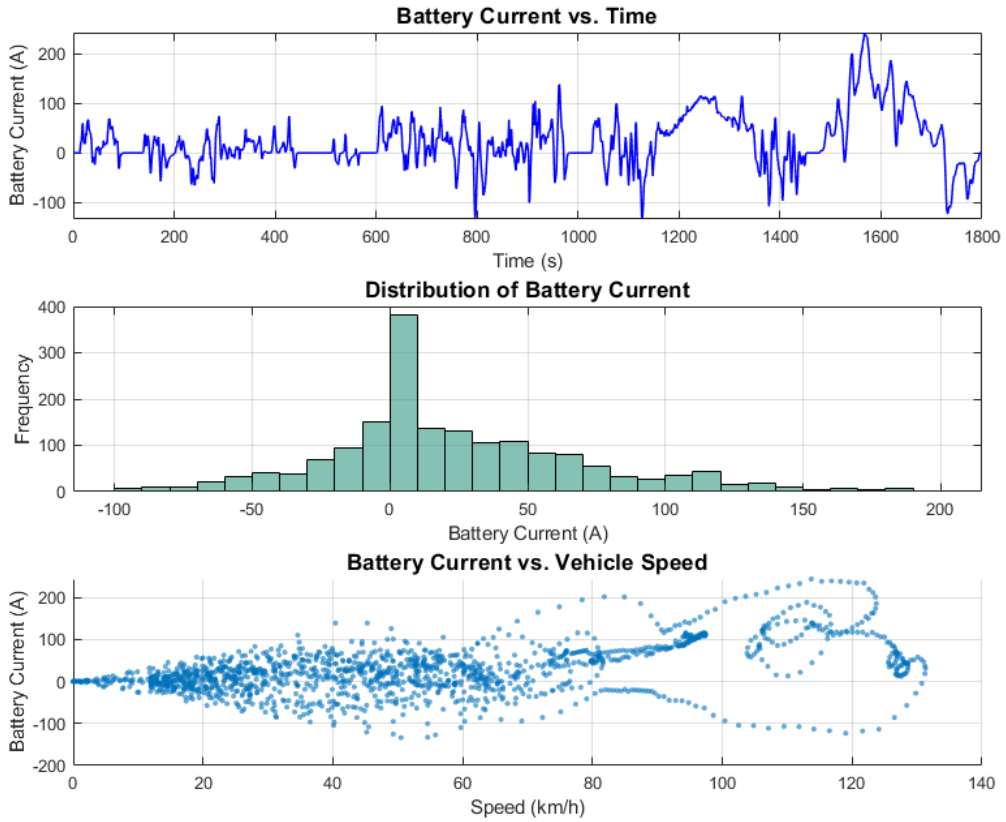

Figure 4: Battery current over the WLTP Class 3B cycle (top), its overall distribution (middle), and its relationship to vehicle speed (bottom)

Middle Plot (Distribution of Current): This histogram in Figure 4 indicates how frequently each current value occurs. The highest bin is around zero (0-10A), indicating steady cruise (driving at constant speed). At this speed, the current is modest yet positive, just enough to overcome air drag and rolling resistance. The graph also clearly demonstrates the distribution of events: discharge spikes rise to +200 A, whereas regeneration spikes fall to -100 A.

Bottom plot (battery current vs. speed): Current values are all over the stop - start place, at low speeds (0–30 km/h). At moderate speeds (40–80 km/h), steady draw is observed because currents cluster is around +50 A. At higher speeds (>100 km/h) the current spikes attain their greatest value and alternately show regen as the vehicle slows.

To compute the battery flow efficiency in percentage, following expression is used:

$$e_{batt} = \frac{C_{sum}}{C_{sum} + D_{sum}} \times 100 \qquad (18)$$

C_{sum} is the current which flows towards the battery during the charging phases, and in that instant the value of current is negative. D_{sum} is the opposite case that means total sum of

current drawn from the battery during acceleration and value of current is positive. Figure 5 illustrates the battery current oscillating between discharge and charge phases, with the cumulative current values employed to assess overall efficiency.

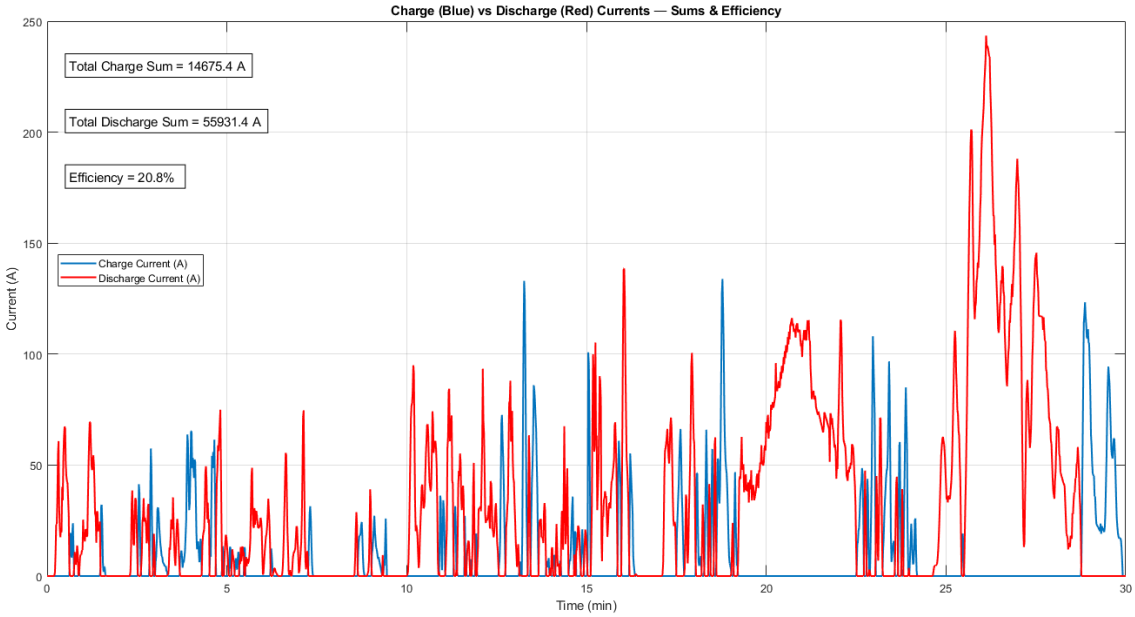

Figure 5: Charge (blue) vs Discharge (red) current over time with total sums and calculated efficiency

3.1.4 Energy Consumption [Wh/km]

To evaluate an EV's performance, accurate energy consumption calculations are required. These estimations form the basis for determining the total system's efficiency. They also enable us to conduct meaningful comparisons between various standardised drive cycles and assess the effectiveness of regenerative braking.

In this thesis, we applied a system like that described by Miri et al. (2020). We utilized this approach to compute instantaneous energy usage using the battery's power data. These figures were then merged to produce a comprehensive trip-level energy balance. The total energy spent over time was calculated by integrating the battery power and dividing by 3600, which translates watt-seconds to watt-hours.

$$E(t) = \sum \frac{P_{batt}(t) \times \Delta t}{3600} \tag{19}$$

The incremental distance $\Delta d(t)$ covered at each step

$$\Delta d(t) = \frac{v(t)}{3600} \tag{20}$$

Energy consumption every second was calculated as

$$EC(t) = \frac{E(t)}{\Delta d(t) + \in} \quad (21)$$

Average energy consumption for a trip were calculated as

$$EC(t) = \frac{E(t)}{\Delta d(t)} \quad (22)$$

Note that ∈ is the smallest value to avoid division by zero when vehicle is not moving. So, in the implementation phase, these above-mentioned equations were applied at 1 Hz to the WLTP Class 3B drive cycle. Positive P_{batt} indicates the traction energy while negative values correspond to regenerative phases.

3.1.5 Regenerative Efficiency Calculation

Regenerative efficiency is one of the main parameters of the extent to which an electric vehicle (EV) can recover the kinetic energy during braking and transfer to the battery. The greater the efficiency of regeneration, the greater the driving range, the less net energy consumed and the greater the overall sustainability profile of the vehicle system.

In extensive literature review, it is observed that most of the studies that are cited used either laboratory-controlled or on-road tests. As the similar facilities were not available, a purely mathematical approaches were used to compute the necessary parameter and then calculate the regenerative braking efficiency.

The kinetic energy lost can be calculated as

$$\Delta\, KE(t) = \frac{1}{2} m[v^2(t_1) - \, v^2(t_2)] \quad (23)$$

Recovered energy is computed as

$$E_{regen}(t) = \, \sum_{I_{batt<0}} \left[V_{pack}(t) \times I_{batt}(t)\right] \quad (24)$$

Regeneration efficiency is the ratio expresses the percentage of lost kinetic energy successfully recovered.

$$\eta_{regen} = \frac{E_{regen}(t)}{\Delta KE(t)} \times 100\% \quad (25)$$

By using these formulas, the regenerative efficiency of WLTP Class 3B drive cycle was computed.

3.1.6 Battery Pack Temperature *T*

The next step in this thesis was to determine the battery temperature. To do this, we used Miri et al.'s (2020) technique to estimate the temperature of the lithium-ion battery pack, updating it every second (at 1 Hz). This technique employs a simplified heat-balance model. It regards the entire battery pack as a single unit (a 'homogeneous thermal mass'), ignoring any minor temperature fluctuations inside the pack. This approach is very 'light' on computing, making it simple for the microcontroller to operate. This makes it ideal for pseudo-real-time analysis when actual lab facilities are unavailable for testing. The model is based on a typical 'one-node' thermal balance, which balances the heat generated by the battery with the heat lost to the environment. This can be expressed as:

$$T_k = T_{k-1} + \frac{I_k^2 R_0 - hA\,(T_{k-1} - T_{amb})}{C_{th}} \qquad (26)$$

Where k indexes represent the per second sample during the whole driving trip of 30 minutes. The parameters in expression are explained in the following Table IV.

Table IV. Parameter used to calculate battery pack temperature

Parameter	Symbol	Value	Unit	Reference
Thermal capacitance	C_{th}	212,000	J/K	BMW Technical Data (mass) + Li-ion calorimetry
Internal resistance	R_0	0.19	Ω	INL BMW i3 pulse resistance data
Heat transfer coefficient	hA	30	W/K	Within 15–60 W/K literature, NREL
Ambient temperature	T_{amb}	25	°C	In literature study, researchers take values in between 25–30 °C
Battery current	I_k	Varies at 1 Hz	A	Already calculated in WLTP profile

3.1.7 State of Charge (SOC) and Remaining Range

Whenever there is a debate about EVs and its parameters, there is something which cannot be ignored and that is state of charge and remaining range because they are the key indicators in EVs, and driver should know how much the battery charge is left and the remaining range that can be covered before recharging is required.

To perform these calculations, two mathematical formulas are needed which are discussed below.

$$SOC(\%) = \frac{E_{cap} - E_{used}}{E_{cap}} \times 100 \qquad (27)$$

Where,

- $SOC(\%)$ represents state of charge in percentage
- E_{cap} is the battery pack's entire useable capacity. In this implementation, the figure is 52 kWh, depending on the parameters of a Renault Zoe R135 (2024 model).
- E_{used} is the total net energy in Wh consumed since the start of the journey.

Calculating remaining range using a basic average energy use can be extremely unpredictable and deceptive for the driver. For example, a single, brief acceleration might cause the average consumption to spike (jump up). This would result in a rapid decline in anticipated range, which is neither practical nor beneficial. This thesis employs a far superior technique known as the Exponential Moving Average (EMA) to provide a more consistent and reliable estimate.

An exponential moving average (EMA) is a smarter sort of average. It operates by giving priority (importance) to the most recent driving data rather than treating all data equally. This causes the average person to adjust his present driving style, but it also smooths out any unexpected, short-term surges. As a result, the driver receives a more steady and reliable range estimate.

$$ema_{new} = (\alpha \times avg_{current}) + ((1 - \alpha) \times ema_{old}) \qquad (28)$$

The complete implementation of the firmware comprised the following crucial details:

- The smoothing factor (α) was set at 0.1.

- For the first 240 seconds of the trip, a reference consumption number of 177 Wh/km was utilized to maintain stability.

Following this first time, the EMA method was used, with a safety floor to ensure that the number did not fall below the reference, avoiding unduly optimistic range projections during periods of abnormally low consumption. The remaining range was estimated by dividing the remaining battery energy by the smoothed consumption value (ema Wh/km):

$$Range_{left}(km) = \frac{E_{rem}}{ema} \quad (29)$$

Where E_{rem} is the remaining energy in the battery, calculated from Eq 27 $E_{cap} - E_{used}$. This EMA-based technique yields a considerably more reasonable and less unpredictable range estimate for the driver.

3.1.8 Battery Temperature Prediction using Edge AI

This section shows how we used regression-based machine learning to anticipate the temperature of the battery pack two minutes in advance. Battery temperature is an important metric in EVs since it influences both efficiency and safety. In case the battery overheats and gets beyond the limit, the battery may perform poorly, have reduced life or even cause death in worst scenarios.

The driver can make safer judgements with the help of forecast of the temperature. Our dataset was developed using various recorded driving excursions and we have FTP72, FTP75, and WLTP driving braking cycles. These records cover a large variation of road circumstances with all the information being recorded at one second (1 Hz). This supplied a diverse range of driving scenarios, giving the model lots of informative data to learn from.

The characteristics of the input which we explained in detail in the preceding section were used to train the model. The label (predicted value) was fixed to the prospective temperature of the battery. This anticipatory power allows to better manage the thermal performance, which can be used to sustain the optimum operation of the battery and leave the driver aware of the future events.

3.1.9 Dataset Preparation, Feature Selection, and Output Label

To train the model, the driving datasets came from the known driving cycles: FTP72, FTP75, and the WLTP brake cycles. These standard driving cycles are used to regulate how EVs perform in various driving conditions like starting, stopping, speeding up, and slowing down. Just like the WLTP Class 3B, the other driving sets chosen for training are already sampled at 1 Hz, so there is no need to change the data. The trips were appended into one csv for easier use in training the model in Edge Impulse. In csv file, whenever a trip ended and new one started, the time column was reset to zero so that every trip started from the same time. This helped to keep data consistent and avoid from any confusion.

The 10 input features were utilized which were responsible in changes the battery temperature during the trip. The order of features remained same as the csv of WLTP Class 3B driving cycle and shown in Table V below.

Table V. Order of features in WLTP Class 3B cycle and in training datasets

No.	Feature
1	Speed
2	Distance
3	Acceleration
4	Road elevation
5	Coasting
6	Battery current
7	Energy consumption
8	Kinetic energy lost
9	Electrical energy recovered
10	Battery pack temperature

The output label needed to be predicted is a battery temperature for future 2 minutes depending on the driving behaviour. At the end of trip, the data became less than 2 minutes still whatever the future data remained was used to train the dataset so that every row was included in the training.

All features were normalized using scikit-learn's Standard Scaler method before training. This method prevents features with high values such as battery current to overcome the small range features like road elevation. No values were missing, so no special cleaning was

required. These selected features provide a complete picture of the EV's thermal, electrical, and driving dynamics. Speed, acceleration, road gradient, and battery current represent power demand, while energy consumption and kinetic energy loss represent heat generation within the battery. Regenerative energy minimises overall energy consumption by recovering some energy during braking, which also indirectly affects battery temperature. The current battery temperature serves as an initial state, from which future temperatures are estimated. This combination gives the model the ability to learn complex relationships that are vital to predict changes in battery temperature.

3.1.10 Model Architecture (Edge Impulse)

The model of this machine learning was trained using Edge Impulse. It is a regression model, or in other words, it is one that is meant to estimate specific number and not a category.

The model is supposed to project 10 driving parameters (like speed, current, etc.) to one continuous output, the temperature of the battery for next 2 minutes. Our major goal was to create a small and fast model, which can be run on the small ESP32-S3 microcontroller directly. This allows the prediction on the edge computing environment in real time without having a cloud server.

Our dataset consists of 10 characteristics that are integrated in the input layer of the model, and they are the speed, distance, acceleration, road gradient, coasting, battery current, energy used, kinetic energy loss, recovered energy and current battery temperature. We normalised all features before training. It is an important measure to avoid letting large quantities of (say) speed take too much power over minor numbers (say) temperature. Normalization also assists the model to train faster and regularly.

The design of the model begins by having a giant concealed layer of 128 neurons that employs the conventional ReLU activation work. To make the model familiar with such complex correlations among all the input characteristics at the onset, we used a big layer.

Then a dropout layer with a 10% drop out was added. Dropout is a strategy that randomly turns off certain neurons while training. his enhances the generalizability of the model, and also avoids overfitting (also known as simply memorizing the training data). This played a critical role in our study, as there was sparse training information. This is ensured by dropout

since it makes sure that the model does not learn the noise but the actual patterns and can then be used to work with the new data.

The architecture of the model is then made smaller. The second hidden layer consists of 64 neurons, and the third layer has 16 neutrons. The downward trend which rose by 128 and then dropped by 16 was not in vain. It is dealing with the complex patterns and thus makes the model light and efficient enough to be able to run with our modest hardware and only the data that is most important is addressed.

Lastly, the final neuron belongs to the output layer whose activity is linear. It allows the model to give any actual number, and this is what we desire to predict a continuous temperature value (say 26.5 o C). Our model was trained during 300 epochs (cycles) using the standard Adam optimizer and learning rate equal to 0.001. The loss measure used was the Mean Squared Error (MSE) that is a commonly used loss measure in regression because it would apportion large errors severely. The reason we picked regression was we wanted to obtain specific temperature to indicate and not a wide range such as low or high which would not be exact. It can be viewed that the regression network used in this work is in Figure 6 which is composed of input layer (10 features), Dense-128 (ReLU), Dense-64, Dense-16 and linear output, (1 value), and Dense-16, Dense-128, and Dropout 0.1.

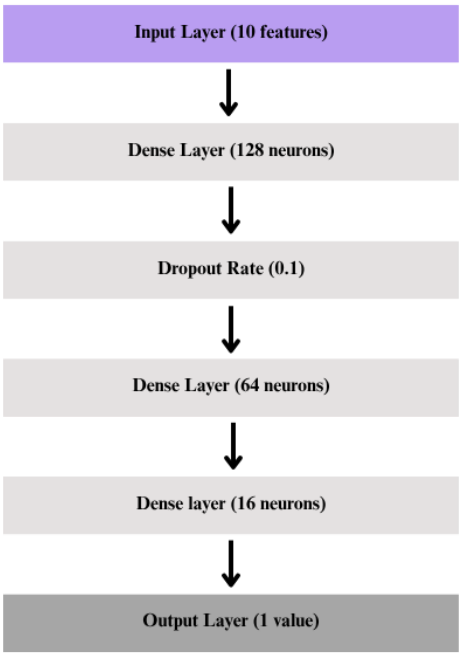

Figure 6: Model architecture showing input, hidden, dropout, and output layers.

This model was particularly well-suited for edge deployment, as it had fewer parameters, a lower computational burden, and ran in real time on low-power devices. Dropout helped to prevent overfitting, and gradual neuron reduction kept the model lightweight and fast. In summary, this architecture was a balanced combination of complexity, accuracy, and practicality. The large initial layer helped learn the relationships between features, the dropout model generalizes, and the regression framework enables accurate and real-time predictions, which were crucial for proactive thermal management of electric vehicles. The final model performance metrics and on-device resource usage are summarized in Table VI and VII.

Table VI. Model performance metrics

Metric	Value
Mean squared error	0.01
Mean absolute error	0.07
Explained variance score	0.99

Table VII. On device performance

On-device performance		
Inferencing time	Peak ram usage	Flash usage
6 ms	2.0 K	51.6 K

Table VI explains that the trained model is very predictive as shown in the following descriptive measures:

- Mean Squared Error (0.01): the difference between the predicted and the measured values of data is small.
- Mean Absolute Error (0.07): the average of the error of the individual is quite low.
- Explained Variance Score (0.99): the model explains 99 % of the variation observed in the data, which is a very good predictive power.

Table VII shows the ability of the model to run on a microcontroller. The Inferencing Time makes predictions in just 6 ms. Peak RAM Usage (2.0 K) shows that the model requires very little memory to run, and Flash Usage (51.6 K) shows that the model size also takes up very little storage space, which is ideal for embedded systems.

3.1.11 Model Testing and Results

The WLTP Class 3B driving cycle is used as the testing dataset to evaluate the accuracy of a model trained in Edge Impulse. The main goal of this phase was to determine the ability of the model to predict the real-world situations. The results showed an accuracy of 95.2% which implies good predictive reliability. Table VIII shows the regression performance of the tiny machine learning model on the WLTP test dataset, which were used to measure forecast accuracy and the variation.

Table VIII. Regression-based performance metrics of the trained model on WLTP test dataset

Metric	Value
Mean squared error	0.17
Mean absolute error	0.09
Explained variance score	0.95

Feature explorer graph in Figure 7 indicates the distribution of the points. The green dots are the correct outcomes of the regression, and the pink dots are where the forecast was not equal to the observed value. Nearly all observations fall on the right regression line hence strengthening model accuracy.

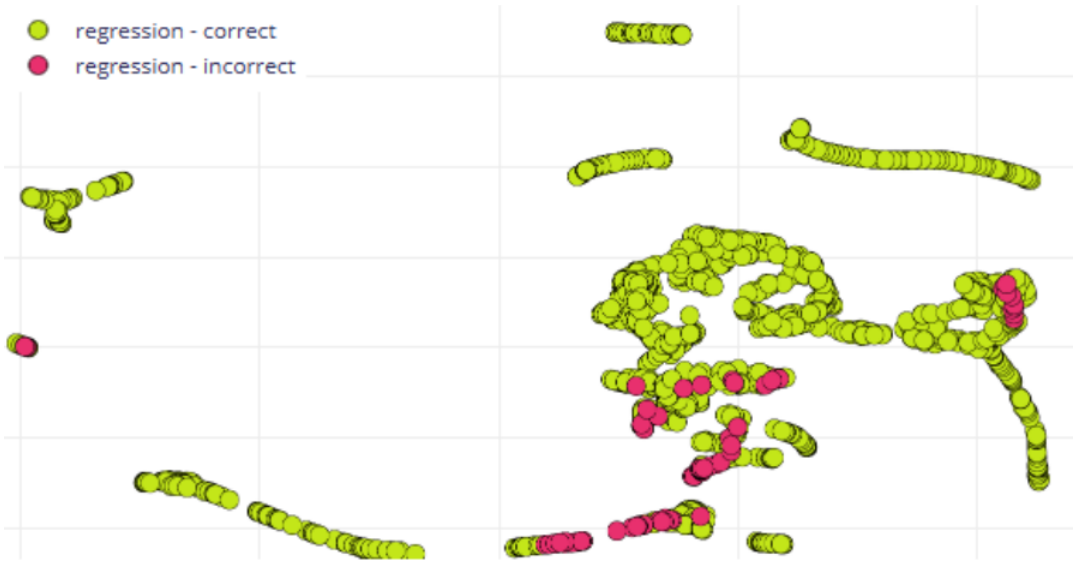

Figure 7: Model testing prediction: Green dots are correct predictions, and pink dots are incorrect ones. The model's accuracy is 95.2%.

This subsection represents the quantitative verification of the TinyML regression model, equivalent to MATLAB-based validation, which was carried out using Edge Impulse's internal testing module.

3.2 Arduino IDE Based Edge AI Implementation

This section explains the full development and deployment of an Edge AI, which is implemented on the ESP32-S3 microcontroller via the Arduino IDE. The device was set to process the WLTP Class 3B drive cycle data set directly at the edge, without requiring any cloud resources. Also, all performance parameters were calculated locally on the microcontroller (edge processing) – energy efficiency, regenerative braking efficiency, coasting percentage and driver behaviour scores. A TinyML model, which was trained in Edge Impulse and imported into the Arduino environment, was used to give real-time battery temperature predictions. The device also processed data into JSON and sent it to a Node-RED dashboard via the MQTT protocol. The dashboard presented trip stats, predictions, and efficiency scores in an interactive format. The Node-RED flow was imported to the FlowFuse public cloud platform, which enabled secure and real-time remote access to live driving analytics from anywhere there is an internet connection. The overall edge-to-cloud architecture of the implemented system is shown in Figure 8.

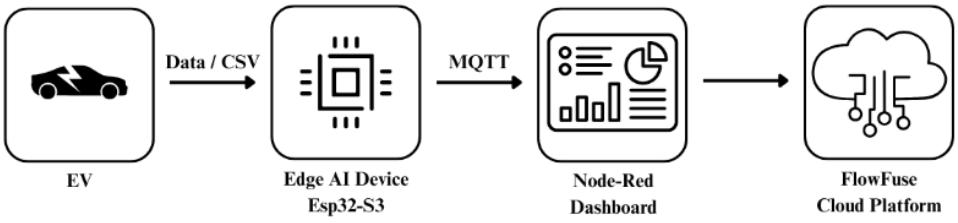

Figure 8: End-to-end architecture of the proposed edge-based EV driving analytics system.

3.2.1 Development Environment and Tools

The ESP32-S3 edge device used via Arduino IDE in implementation phase. Arduino IDE was chosen as it has a large community support and supports ESP32 based boards. The hardware choices were limited by the tight resource constraints inherent in microcontroller-class devices. The ESP32-S3 has 512 kB SRAM and 8 MB flash, requiring a compact model

design and efficient firmware processes. To ensure real-time performance, the TinyML regression network's forward inferences must be completed within a single 1 Hz cycle (≤ 10 ms). As a result, layer widths and dropout rates were set to balance accuracy against time and memory margins. This constraint-driven design demonstrates how computational feasibility directly influenced the model structure and deployment feasibility at the network edge. Few libraries are important to put together the base of project:

- WiFi.h: The library was used to get the ESP32-S3 on to a wireless network. This library takes vehicle of the Wi-Fi connection set up and the device's ability to send data to remote servers.
- PubSubClient.h: The library was used to integrate this to enable MQTT communication between the ESP32-S3 and the MQTT broker. This enables the device to send out processed data and to receive any control messages.
- LittleFS.h: Used for local storage of CSV files in the ESP32-S3's flash memory, allowing device to process dataset during simulation without requiring live data feed.
- my_project_inferencing.h: This file/model was generated when exporting from Edge Impulse machine learning model as an Arduino library. It includes all the functions and model data required for running TinyML inference on the ESP32-S3.

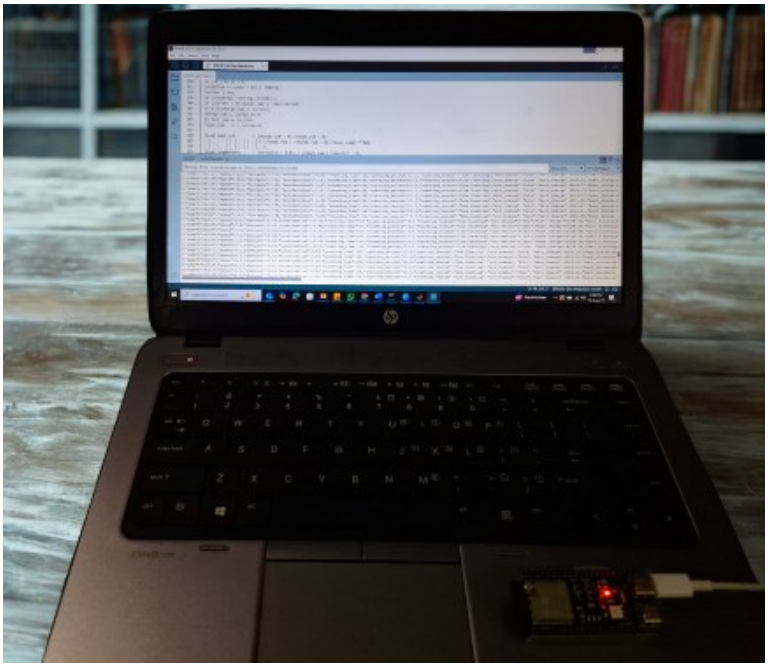

Figure 9: ESP32-S3 microcontroller setup with Arduino IDE for pseudo real-time EV trip data processing.

3.2.2 Offline Dataset Processing and Pseudo Real-time Edge Calculations

This subsection explains what the ESP32-S3 did with the WLTP Class 3B 1 Hz data set when processed locally and transformed into trip reports in pseudo real time. The data was stored in the device's flash using LittleFS read through at run time. Every row has time, speed, cumulative distance, acceleration, road slope, a coasting flag, battery current, instantaneous energy, kinetic energy loss, regenerated energy, and battery temperature. At the edge, the firmware maintains running totals and at about 1 Hz is putting out the following metrics:

- Trip distance was a sum of all distance at 1 Hz calculated from speed.
- Coasting percentage was the ratio of accumulated coasting time to total trip time which shows how much of the trip, driver spent not using the accelerator.
- Battery flow efficiency was the ratio of the magnitude of charge to the total of charge and discharge currents to show how well current flow is managed.
- Average energy consumption (Wh/km) was the cumulative energy divided by distance.
- Regeneration efficiency was the total of recovered energy to kinetic energy losses which in turn indicates the degree of braking energy recovery.
- Thermal status was derived from the measured battery temperature.

The ESP32-S3 translated raw trip measurements into qualitative statuses according to pre-determined thresholds. Such thresholds enable a driver and analyst to interpret results without having to look at raw numerical values. The status of all the metrics which was employed to give a weighted driver scoring model to provide an overview of the overall performance is shown in Table IX.

Table IX. Performance Metrics, Thresholds, and Scoring Weights

Metric	Status threshold	Score Weight	Remarks
Energy Consumption (Wh/km)	Efficient: < 140 Moderate: 140–180 Inefficient: > 180	35%	Main efficiency indicator. Lower values are better.
Regeneration Efficiency (%)	Good: > 70% Moderate: 40–70% Low: < 40%	25%	Measures braking energy recovery.
Battery Current Efficiency (%)	High: > 60% Moderate: 30–60% Low: < 30%	15%	Reflects current flow management.
Coasting Percentage (%)	High: > 15% Moderate: 8–15% Low: < 8%	10%	Indicates smooth driving with less acceleration.
Battery Temperature (°C)	Optimal: 20–35°C Cold: < 20°C Warm: > 35°C	15%	Thermal performance; affects efficiency and lifespan.

The final driver score was figured as the weighted sum of each metric's score. Scores are classified into:

- Excellent: ≥ 80
- Moderate: 50–79
- Inefficient: < 50

3.2.3 Node-RED Dashboard and Public Cloud Deployment (FlowFuse)

The visualization layer was built in Node-Red for local deployment as well as in cloud deployment on FlowFuse. The edge device pushed telemetry over MQTT via HiveMQ broker. A MQTT In node subscribes to topic trip/data and the data was passed to a JSON node to be parsed. The result of the parsing was then passed on to different UI widgets without further processing since all the calculations and status mapping had been done on the ESP32-S3 prior to sending. As illustrated in Figure 9, the ESP32-S3 runs the whole

firmware inthe Arduino IDE, creating pseudo real driving metrics and uploading the results as JSON data.

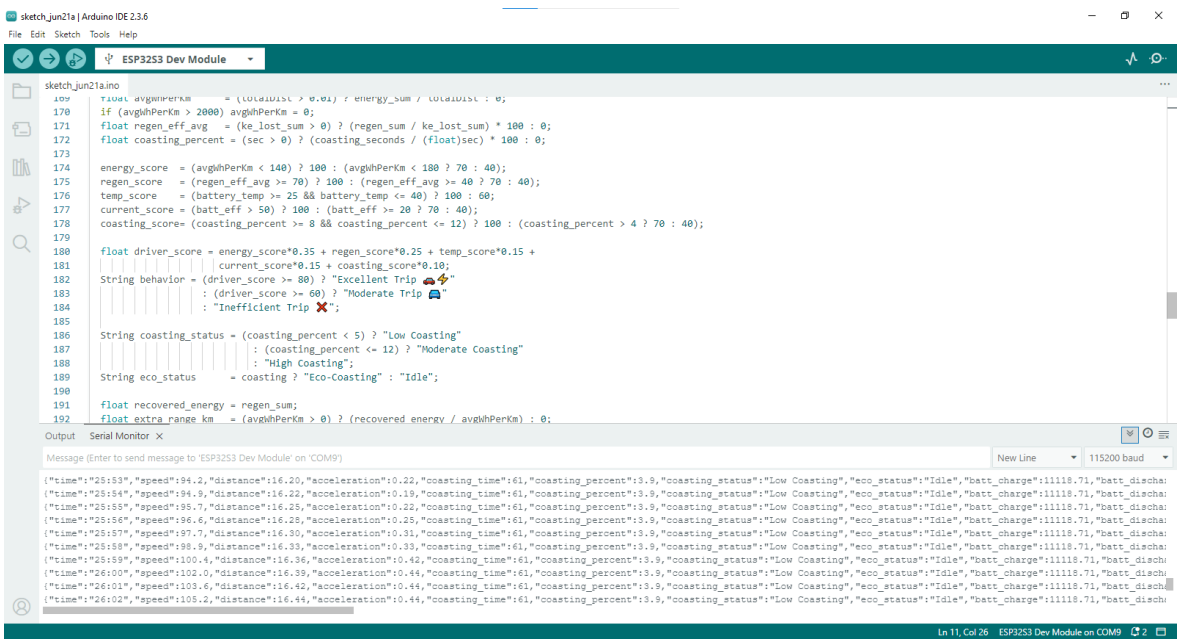

Figure 9: ESP32-S3 Serial Monitor output showing computed trip metrics and status values after on-device edge processing

The dashboard presents data in a clear and interactive format:

- Gauges which include key real time metrics like battery flow, battery temperature, energy consumption, coasting, regenerative efficiency and driver score.
- Time series charts which display energy consumption (KWh/km), battery current flow, and regeneration efficiency over time thus enabling trend analysis per trip.
- Status indicators which give a quick picture of performance in terms of energy use, regeneration quality, current stability, coasting behavior and thermal conditions.

Color coding was used to apply based on status strings in the JSON payload. For example status *Efficient*, *Moderate*, and *Inefficient* have mapped to green, grey, and red respectively. This is done in a way that the dashboard's visual elements always align with the edge device's decision making which removes the risk of different threshold issues between the device and the UI. As seen in Figure 9, the ESP32-S3 runs the whole firmware within the Arduino IDE, creating real driving metrics and uploading the results as JSON data.

As shown in Figure 10, the Node-RED dashboard flow subscribes to the MQTT topic "trip/data" and delivers real-time values to widgets.

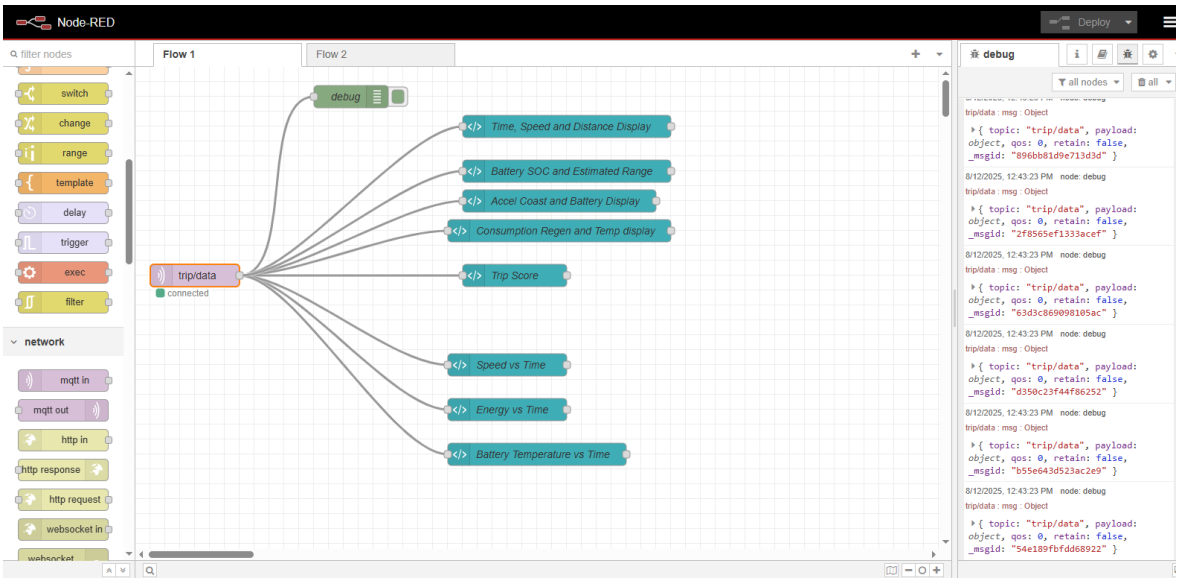

Figure 10: Node-RED flow for visualizing processed trip metrics from the ESP32-S3

For public access, the Node-RED flow which includes its MQTT setup, JSON parser, and UI design was exported out of the local environment and imported into FlowFuse. The FlowFuse instance which in turn connects to the same HiveMQ broker enables real time cloud-based visualization which does not require additional tunneling or VPNs. The FlowFuse dashboard was published via its public share which generates a live read only URL that allows for remote monitoring from any web browser.

To maintain reliability the MQTT nodes were configured with keep alive intervals, auto reconnection, and broker credentials stored securely in FlowFuse environment variables. This architecture was applied to both local and public dashboards to get identical low latency data streams directly from the edge device.

4 Results and Analysis

The dashboard reports full trip performance which is designed to put out key driving and battery related parameters in a clean and organized way which allows for quick study of the main results. The interface is fragmented into many info tabs which each look at a different aspect of the trip.

Trip metrics like time, distance travel, vehicle speed, battery state of charge (SOC), estimated range left, acceleration trends, time spent coasting, battery current, energy use, regenerative efficiency, and battery temperature are included. Trip efficiency scores are calculated, which consider many performance variables to present an overall picture of how well the vehicle is driven.

All information is displayed via dashboard cards for simple interpretation. Color-coded status indicators to note if the values are in the optimal range or need improvements. This layout is to present all the important parameters at a glance with more in-depth data available in the sub sections.

Table X summarizes the major WLTP Class 3B trip data, including distance, consumption, regeneration, coasting share, thermal behavior, and on-device timing. This data gives a short numerical picture of the whole trip prior to the complete dashboard analysis. Figure 11 shows the complete Node-RED dashboard with all trip indicators.

Table X: WLTP Class 3B — Trip Metrics (Summary)

Metric	**Value**	**Note**
Trip distance	23.27 km	From WLTP Class 3B replay
Net energy used	4.20 kWh	Matches with MATLAB plot
Average energy consumption	180.5 – 180.6 Wh/km	Consistent across dashboard & MATLAB
Coasting time	61 s (3.4%)	From Fig. 14
Regenerative efficiency (end of trip)	61.7 %	From Info Tab/section text.
Recovered range (regen)	5.43 km	From Info Tab/results
On-device loop time (compute +inference)	2.15 ms (mean)	Telemetry timing table.

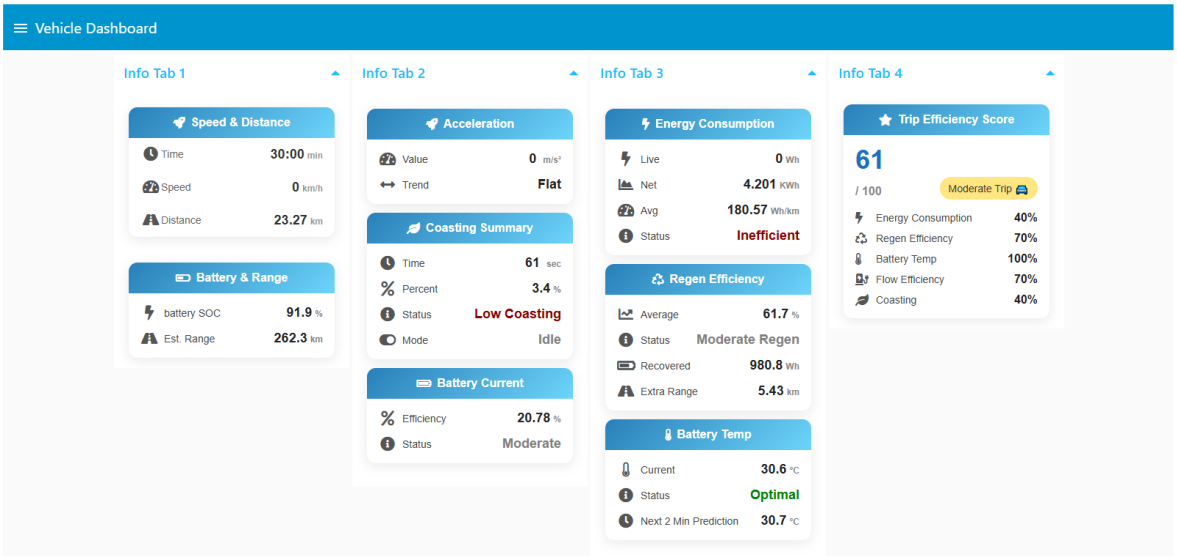

Figure 11: Overall layout of the vehicle performance dashboard

4.1 Info Tab 1 – Vehicle Status

The Info tab 1 gives an instant overview of the fundamental driving parameters of the trip, such as Speed & Distance and Battery & Range. Four snapshots of the snapshots of the parameters taken at various stages of the journey show the change in the parameters between the beginning and the end of the journey.

The following figure demonstrates four major snapshots of the Info Tab 1 vehicled at various points of the trip. These are visual images that show the development of speed, distance, battery state of charge (SOC), and estimated range, which are informative for the driver during the whole drive.

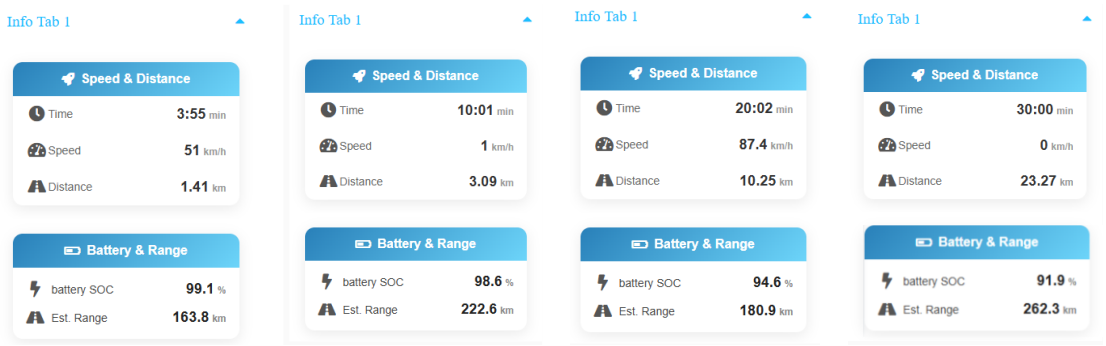

Figure 12: Four Info Tab 1 snapshots showing vehicle status at different trip stages

In Figure 12, at 3:55 the vehicle speed is 51 km/h after covering 1.41 km, at this stage the battery SOC is 99.1% and an estimated range of 163.8 km. At 30:00, when the trip finished

with a total distance of 23.27 km; at that point the SOC is 91.9% and the estimated range had reduced to 262.3 km, indicating the remaining range after the trip. This is the exact distance shown in figure 2, MATLAB plot in Chapter 3.

4.2 Info Tab 2 – Coasting Summary & Battery Current

The Driving behavior section in Info Tab 2 presents two noteworthy indicators of energy efficiency, which are Coasting Summary and Battery Current. These parameters allow the driver to classify how the driving style affects energy consumption and vehicle efficiency during a trip. The images in Figure 13 provide examples of the values recorded at different times.

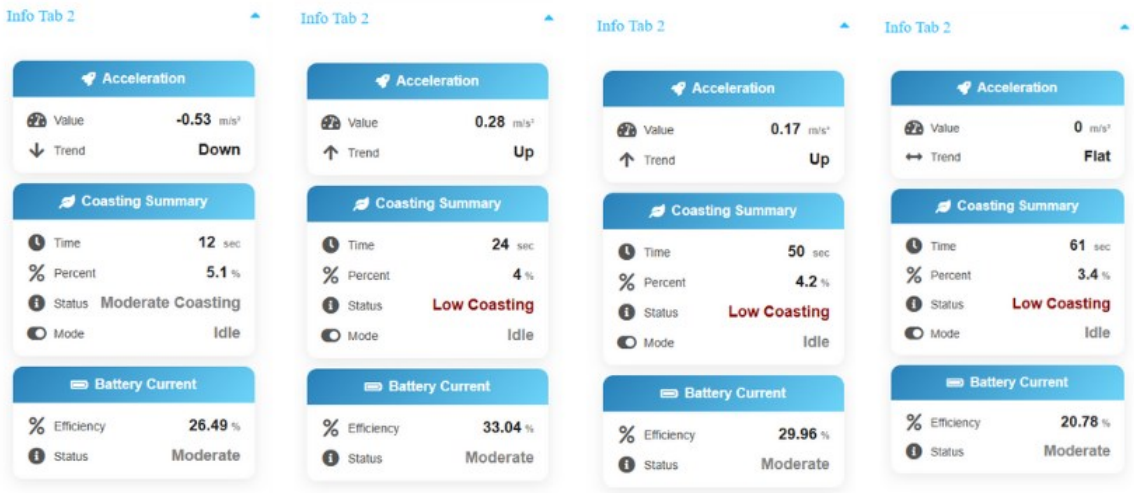

Figure 13: Info Tab 2 showing coasting and battery current efficiency at different trip stages

The Coasting Summary vehicled depicts how long the vehicle has been moved without pressing the accelerator and how often this has happened.

- Time indicates the total number of seconds the vehicle is coasting within the trip stage shown.
- Percent represents part of the trip time spent coasting.
- Status defines the coasting performance as Low, Moderate, or High, depending on the coasting duration by the logic thresholds.
- Mode indicates the vehicle's operational state during coasting (in these snapshots, shown as Idle).

In the WLTP Class 3B run, the end-of-trip coasting share was 3.4% (about 61 seconds of zero-torque motion in 30 minutes). This interval results in around 72 Wh less net energy usage for the journey (≈ 1.7% of the 4.2 kWh total), indicating a modest but significant range increase (about 5-6 km on a 300 km EV). Coasting phases also reduce peak discharge currents, reduce internal heating and keep the pack below 31 °C, which is good for long-term battery life. These effects demonstrate why larger coasting percentages are often associated with increased travel efficiency.

For instance, at 12 seconds of coasting time, the percentage is 5.1%, categorized as Moderate Coasting. Later, at the end of trip, the percentage is seen as only 3.4%, categorized as Low Coasting with total coasting time 61 seconds. The total coasting time can also be verified from Figure 14 (MATLAB plot which indicates total coasting time). Low coasting percentages imply that the vehicle has been subjected to more frequent acceleration or braking, which can decrease energy efficiency. Higher percentage indicates that less energy has been wasted due to smooth driving and better conservation of energy.

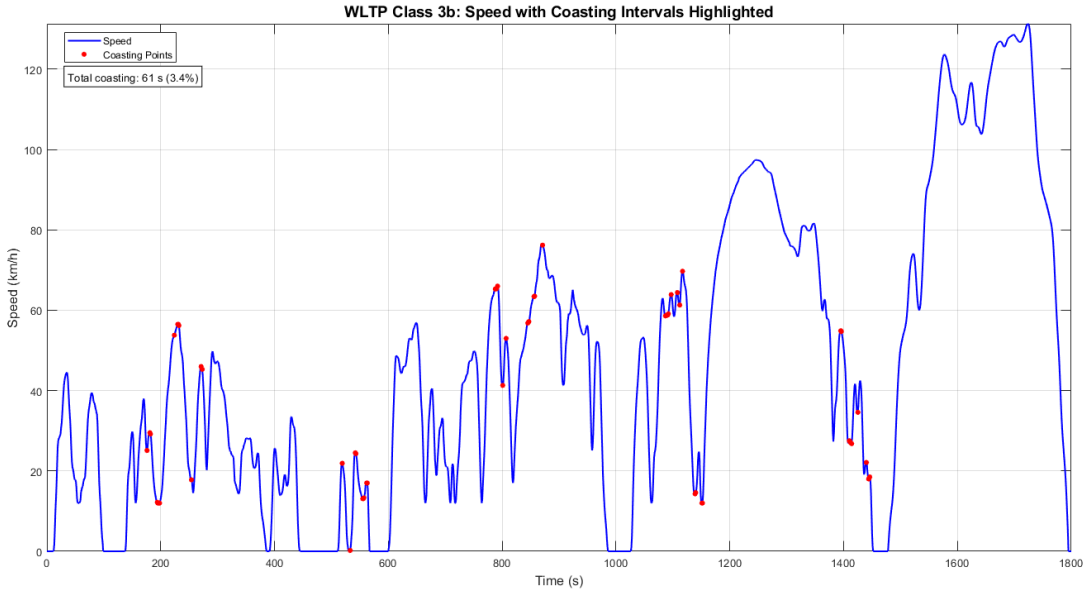

Figure 14: WLTP Class 3B speed trace with coasting intervals and showing total coasting of 61 s (3.4 %).

Below the coasting vehicled, battery current vehicled displays the efficiency of battery current flow during the entire trip:

- Efficiency in percentage directs the effectiveness of current draw to the support of the propulsion and auxiliary systems to the energy losses.
- Status converts the percentage into the level of performance.

As per the Figure 13, the efficiency values started from 33.04% and ended at 20.78% during the trip with Moderate status. The higher the percentages, the more stable the driving is at the optimal speeds, whereas lower percentages might be at the time of frequent acceleration or increased loads.

Together, these measures give the driver feedback in real-time on how smooth the driving is being done. Coasting Summary can also be utilized to track and improve range by encouraging slower acceleration and greater coasting. Battery current efficiency provides the driver with information on how well the battery is being used and adjustments needed in driving style to optimize performance and energy savings.

4.3 Info Tab 3 – Energy Usage and Battery Health Indicators

The Info Tab 3 has 3 important parameters of EV performance: Energy Consumption, Regenerative Efficiency, and Battery Temperature. These parameters are a straight indication of how good the vehicle is utilizing the stored energy, how much it is being recovered during driving and whether the battery is kept within safe operating conditions. These metrics play a pivotal role in the drivers in extending the range, avoiding energy wastage and protect the battery lifecycle.

Energy Consumption vehicled (shown in Figure 15) displays the current and cumulative energy consumption. Let’s dive in the details of vehicled shortly:

- Live (Wh): The amount of energy usage at every second during the trip.
- Net (kWh): The total net energy amount which was used during the trip.
- Avg (Wh/km): Specifies the average consumption per km driven.
- Status: Shows the status of the level of consumption to be Efficient, Moderate or Inefficient based on the criteria set at edge device.

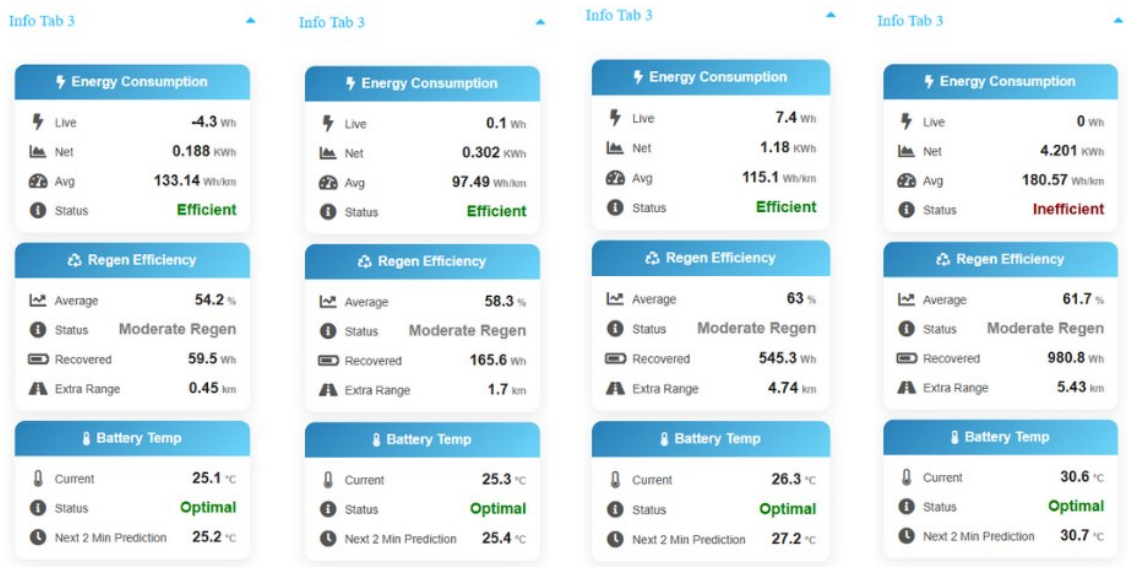

Figure 15: Info Tab 3 showing energy consumption, regenerative efficiency, and battery temperature at different trip stages

In Figure 15, Efficient status is observed where average values are between 97 to 133 Wh/km in time 3:55 to 20:02 (from left to right). The status changed to Inefficient when trip was completed indicates 180.57 Wh/km (most right) shows aggressive driving and high-speed driving actions. Observing this indicator will allow the driver to change the speed and driving style to maintain the energy consumption at a minimum to increase the range per charge.

In MATLAB plot of Figure 16, blue line represents instantaneous energy use in Wh for whole trip travelled, and the orange curve, whose units are shown on the right-hand axis, represents cumulative net energy use over time. The total net energy divided by the distance of the trip gives net energy consumption of the whole journey of 180.6 Wh/km. The graph represents the net energy consumption which is 4.2KWh.

The Regenerative Efficiency vehicled tells the level at which the vehicle can convert the braking energy back to stored energy during the braking or deceleration process:

- Average (%): The measure of the regeneration efficiency of the segment.
- Recovered (Wh): The amount of energy recovered from driver braking action.
- Extra Range (km): The equivalent driving range that is gained using regeneration.
- Status: Categorizes performance as either Moderate Regen or higher.

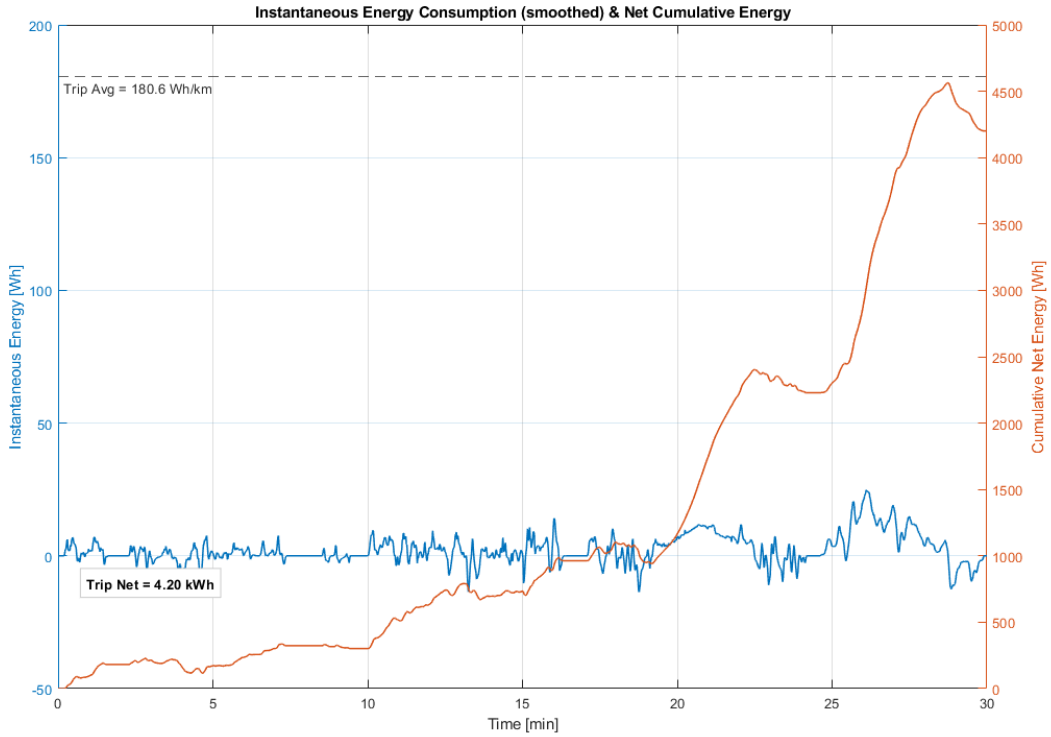

Figure 16: WLTP driving cycle analysis trip average and net energy consumption trends.

The moderate regen performance achieved an efficiency from 54.2% and 63%. The maximum range obtained in these snapshots is 5.43 km, which driver recovered from braking. At the end of trip, the regen efficiency is recorded as 61.7% with overall moderate status.

The higher the percentage of regeneration, the less the battery is drained on a net basis. Drivers can enhance regenerative braking efficiency by predicting traffic, minimizing hard braking, and coasting longer before braking.

The battery temperature vehicled is a key indicator of battery performance:

- Current (°C): Present battery pack temperature at 1 Hz.
- Status: Reports as optimal when in the safe temperature range.
- Next 2 Min Prediction (°C): Reports temperature trends which will alert the driver to possible overheating.

Temperatures stayed in the range of 25.1°C to 30.6°C (in Figure 17) and reported as optimal. Keeping the battery at optimal temperature improves efficiency, prevents thermal breakdown, and also avoids power restrictions which present with overheating. By using

these values in WLTP Class 3B trip, MATLAB graph is plotted to visualize the temperature in whole trip.

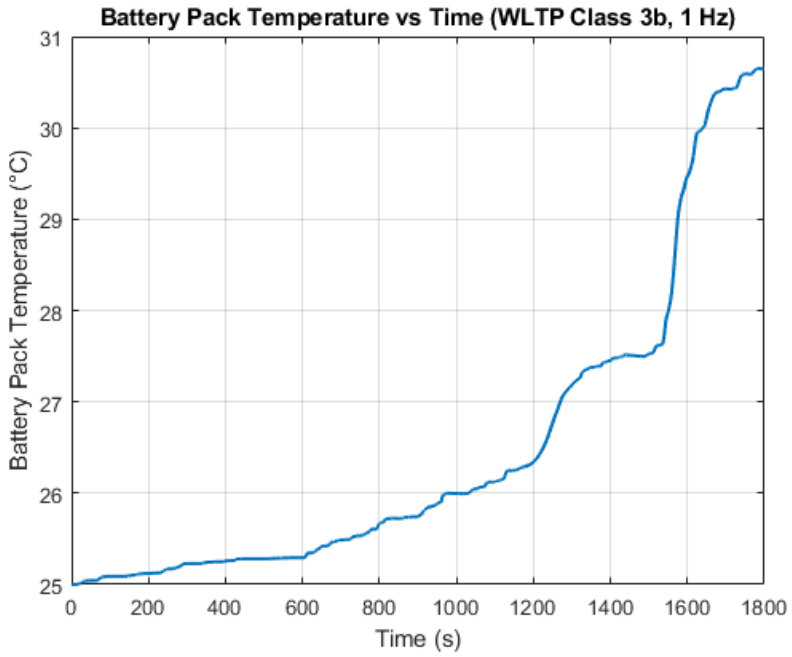

Figure 17: WLTP Class 3B battery temperature vs. time for complete trip

Figure 18 compares the actual battery temperature to the model's 2 minutes ahead estimate during the WLTP Class 3B trip. The two curves were quite close for the duration of 30 minutes, showing accurate short-term forecasting. In the magnified panels, Base 15 minutes indicates that the starting point was 15:00 and the value was confirmed 2 minutes later at 17:00: the model predicted 26.1 ◦C, while actual temperature was 26.0 ◦C (0.1 ◦C difference). The same applies for Base 20 minutes, where the start time was 20:00 and the verification time was 22:00: the model predicted 27.5 ◦C and the actual temperature was 27.3 ◦C (a 0.2 ◦C difference). These little differences suggest good agreement.

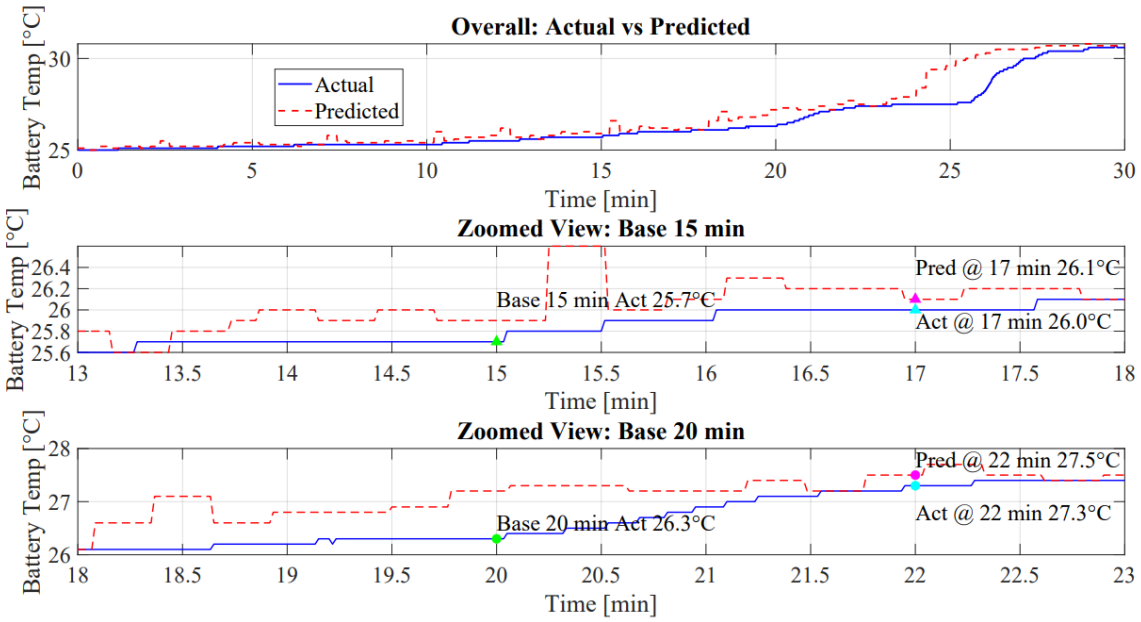

Figure 18: Actual and predicted trends of temperature with comparisons at aggregate and selected times.

4.4 Info Tab 4 – Trip Efficiency Score

The trip efficiency scorevehicled is a full story of the vehicle's performance over the course of the trip. It combines performance-based metrics into one complete score which in turn gives an easy assessment of the total driving efficiency.

In the beginning and middle of the trip the scores ranged between 82 and 85 which put it in the excellent trip category, very high results across the board in the trip efficiency score. At the end of the trip the overall score went down to 61 as shown in Figure 19 which is in the moderate trip category. The primary cause of the drop is in energy consumption efficiency which caused down overall trip score.

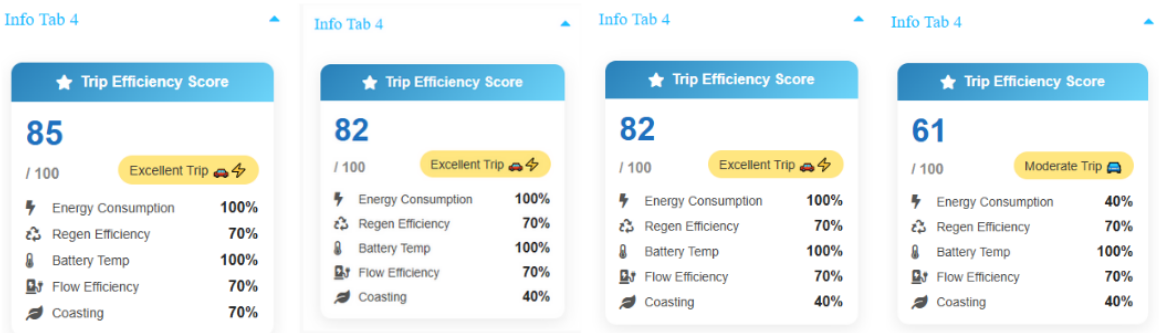

Figure 19: Info Tab 4 showing trip efficiency score and parameter breakdown at different stages

In this trip, it is noticed that extra highway phase of WLTP Class 3B greatly increased energy consumption, which in turn brought down the overall score. By looking at the score distribution, the driver can see which area has the greatest impact and where the improvement is needed for future drives, for instance, keeping speed stable, improving coasting, and better use of regenerative braking.

The ESP32-S3 firmware is lightweight and maintains lossless 1 Hz telemetry throughout the trip; comprehensive data is presented in Tables XI & XII.

Table XI. On-device resource footprint (ESP32-S3)

Flash image size	≈ 1.00 MB
RAM (globals/static)	≈ 50.8 kB

Table XII. Telemetry delivery integrity and timing (1 Hz, WLTP Class 3B)

Quantity	**Value**	**Notes**
Delivery (received/missed)	1,801/0	ratio = 1.00
Loop time (mean/p95/max)	2.15/2.36/3.24 ms	on-device compute + inference
1 Hz tick (mean/p95/max)	61.9/67.9/69.1 ms	scheduler load

As confirmation, the Node-RED debug log in Figure 20 verifies 100% end-to end delivery for the 30 min WLTP Class-3B run (received = 1,801, missed = 0). The loop time values reveal that each compute + inference cycle ends in ∼2–3 ms, with 95% finishing in < 2.36 ms. The 1 Hz tick times indicate that the scheduler uses approximately 62 ms per second, with 95% of the time falling within 68 ms and the worst-case scenario being approximately 69 ms. These results demonstrate that the ESP32-S3 has enough headroom for reliable, lossless 1 Hz streaming and real-time analytics.

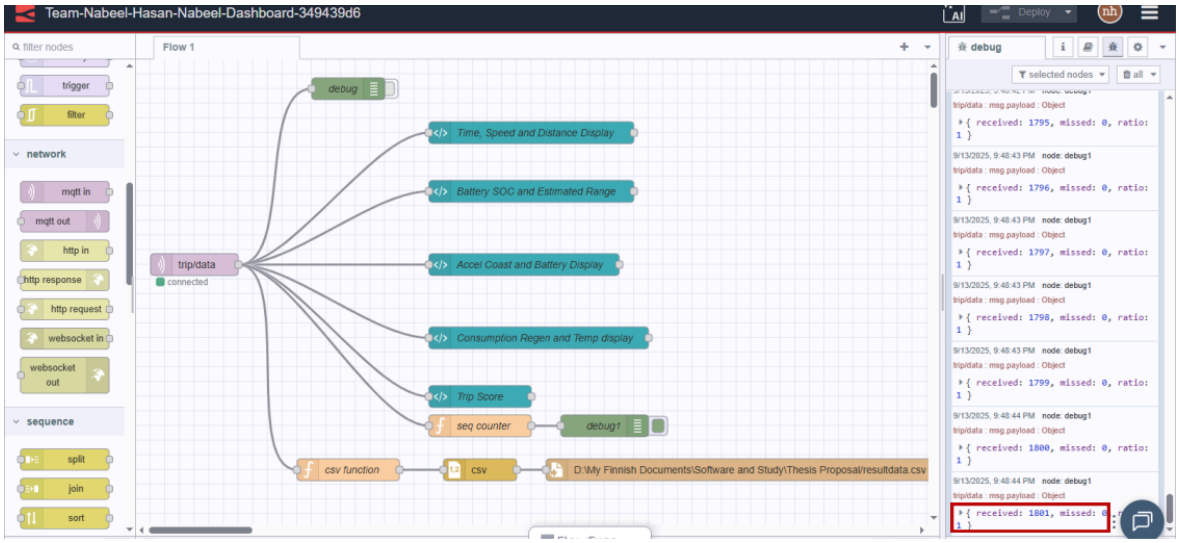

Figure 20: Validation of 1 Hz telemetry integrity in Node-RED.

4.5 Comparative Benchmark: Edge vs Cloud Inference

To get a better idea of how well the ESP32-S3 TinyML model works on devices, it was compared in theory to common cloud-based and offline MATLAB execution paths that have been reported in recent research. The ESP32-S3 completed a single regression inference in 6 ms, which can be considered as fastest for a 1 Hz rate.

Depending on the quality of the connection, the load on the broker, and the distance the data must travel, cloud-native MQTT inference workflows usually have network round-trip latencies of 150–300 ms (Singh et al., 2023; Pamadi & Singh, 2025; Andriulo et al., 2024). This latency is 25 to 50 times longer than local inference and may exceed the real-time budget when it experiences a congestion.

The analysis reveals that edge execution is the best option for real-time thermal analytics in EV scenarios since it is constant, consumes minimal power, and does not need an internet connection.

Table XIII. Comparative benchmark of inference latency and deploy ability

Platform	Typical inference time	Connectivity need	Deployability	Remarks
ESP32-S3 (Edge)	≈ 6 ms (measured)	None	On-board MCU	Deterministic; meets 1 Hz budget
Cloud (MQTT server)	≈ 150–300 ms (literature range)	WAN required	Dependent on network	Variable latency; privacy concerns

4.6 Feature Sensitivity and Interdependence

There is a strong relationship between the 5 metrics used to depict the combined electrical-thermal behaviour. The numerical stability and physical significance of the results are compromised when any one of them is removed.

- If the battery current (A) is ignored, the heating effect disappears, and the relationship between driver behaviour and heat production lessens.
- The temperature of the battery (°C) gives thermal feedback. excluding it, the model is unable to illustrate the patterns of self-heating and cooling that determine how efficient it is and how far it can go.
- Energy consumption (Wh/km) is the total amount of power that flows over a certain distance. There is no comprehensive view of how driving efficiency affects range without it.
- Regenerative braking (%) records the instances when energy is recovered by monitoring the negative current. When it is removed, the overall energy balance is deactivated, and the cooling phases that occur with low load are not visible.
- The coasting proportion (%) reflects windows which consume no power. Ignoring this important metric, the system will believe the EV is still under traction load and more energy is being used than it is.

These relationships confirm that deleting someone upsets power-heat computation and makes it more difficult to anticipate the outcomes.

5 Conclusion

The main aim of the research was to develop a holistic and workable system. This system evaluates the workability of an electric vehicle as well as the behaviour of a driver. We used WLTP Class 3B data as our major source of reference. All the necessary parameters were determined with the help of mathematical models based on this data. This was done with only the speed of the dataset which was 1 Hz, and we did not even need costly laboratory or road testing. To analyze performance of different settings even more in-depth, we have drawn MATLAB graphs.

In the real implementation, we made use of ESP32-S3 microcontroller. The Arduino IDE was programmed with optimised code (firmware) that runs all computation on the device. This comprised obtaining averages, percentage estimates, measuring recovered energy, and determining the range estimate. We used these computations to determine the performance and status of each statistic. These were then added to get the final driver score. This score is shown on a dashboard, which provides the driver with feedback to help them improve their driving.

We trained a machine learning model with Edge Impulse. This model was then installed on the ESP32-S3, transforming the device into a full Edge AI system. We also created a dashboard in Node-RED and put it into the FlowFuse public cloud, allowing any user to view live data.

All these objectives were effectively met. Consequently, we developed an autonomous, low-cost, and effective solution. This device provides drivers with real-time information and analysis on their driving habits, allowing them to save energy.

This study used a narrow set of data of logged journeys (WLTP brake cycles, FTP-72, FTP-75), therefore, the framework may exhibit more error in new traffic, weather, and driver settings. A thermal model is a one-node lumped model with non-varying parameters which merely model pack-specific effects, airflow and ambient variations to some approximate degree. Our testing was done on pseudo real-time replay rather than live tests on vehicles and we did not include an ablation study or estimates of uncertainty.

Future work may be able to verify the pipeline with a test vehicle that has CAN-based measurements of speed, current, voltage, state of charge, and battery temperatures; can better train and test the evaluation to cover a broader drive cycle (US06, HWFET, Artemis, WLTP Class 1/3A) and more diverse environmental conditions (hot/cold, gradients); and refine the thermal model by calibrating it to vehicle-specific parameters and may use a two-node version to provide higher accuracy.

Acknowledgments

We used AI (specifically OpenAI ChatGPT) for writing some of the text in the Methods, Results, and Conclusion sections. All AI-generated content was reviewed and verified by the supervisors for accuracy and appropriateness.

References

1. Andriulo, F.C., Fiore, M., Mongiello, M., Traversa, E. and Zizzo, V., 2024. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics*, 11(4), p.71. Available at: https://doi.org/10.3390/informatics11040071
2. BatteryDesign.net, 2022. Specific heat capacity of lithium-ion cells. BatteryDesign.net. batterydesign.net/specific-heat-capacity-of-lithium-ion-cells/
3. Boretti, A.: Comparison of regenerative braking efficiencies of MY2012 and MY2013 Nissan Leaf. *International Journal of Engineering and Technology Innovation* 6(3), 214–224 (2016). https://ojs.imeti.org/index.php/IJETI/article/view/152
4. Braun, A., Rid, W.: Assessing driving pattern factors for the specific energy use of electric vehicles: A factor analysis approach from case study data of the Mitsubishi i–MiEV minivehicle. *Transp. Res. Part D* 58, 225–238 (2018). https://doi.org/10.1016/j.trd.2017.11.011
5. Chou, K.S.; Wong, K.L.; Aguiari, D.; Tse, R.; Tang, S.-K.; Pau, G.: Recognition of Driving Behavior in Electric Vehicle's Li-Ion Battery Aging. *Appl. Sci.* 2023, 13, 5608. https://doi.org/10.3390/app13095608
6. Dong, S., Wang, Z., Li, J., Li, J., He, Q., Yang, Y.: Task offloading strategies for edge systems—A survey. *Computer Communications* 224, 1–28 (2024). https: //doi.org/10.1016/j.comcom.2023.12.004
7. European Commission (2025) *Apply AI Strategy*. Communication from the Commission to the European Parliament and the Council, COM(2025) 723 final, Brussels. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52025DC0723
8. EVspecs: Renault Zoe R135 (2012–2024) – tech specs and prices. EVspecs.org https://www.evspecs.org/tech-specs/renault/zoe/r135
9. Geng, C., Ning, D., Guo, L., Xue, Q., Mei, S.: Simulation research on regenerative braking control strategy of hybrid electric vehicle. *Energies* 14(8), 2202 (2021). https: //doi.org/10.3390/en14082202

10. Heydari, S., Mirjalili, V., Jamshidi, P., Aminifar, A.: Tiny Machine Learning and On-Device Inference—A Survey of Applications, Challenges, and Future Directions. *Sensors* 25(10), 3191 (2025). https://doi.org/10.3390/s25103191
11. Idaho National Laboratory: BEV Battery Testing Results: 2014 BMW i3 EV – VIN 5655. INL/MIS-14-31587 (2015). https://avt.inl.gov/sites/default/files/pdf/fsev/batteryi5655.pdf
12. Khan, K. (2025) 'What Is TinyML? An Introduction to Tiny Machine Learning', *Talent500 Blog*, 13 November. Available at: https://talent500.com/blog/what-is-tinyml-introduction/
13. Lee, G., Song, J., Han, J., Lim, Y., Park, S.: Study on energy consumption characteristics of passenger electric vehicle according to the regenerative braking stages during real-world driving conditions. *Energy* 283, 128745 (2023). https://doi.org/10.1016/j.energy.2023.128745
14. Li, Z., Shi, Z., Gao, J., Xi, J.: Research on regenerative braking control strategy for single-pedal pure electric commercial vehicles. *World Electric Vehicle Journal* 14(8), 229 (2023). https://doi.org/10.3390/wevj14080229
15. Liu, K., Yamamoto, T., Morikawa, T.: Impact of road gradient on energy consumption of electric vehicles. *Transp. Res. Part D* 54, 74–81 (2017). https://doi.org/10. 1016/j.trd.2017.05.005
16. Lu, M.: Research on New Energy Vehicle Intelligent Control Kinetic Energy Recovery Technology. In: El-Hashash, A. et al. (eds.) *Proceedings of IEIT 2022, Advances in Computer Science Research*, vol. 100, pp. 1015–1020 (2023). https://doi.org/10.2991/978-94-6463-058-9_159
17. Manoj, B.S., Puneeth, Y. and Yuvaraj, S., 2023. Automated vehicle braking system and driver health monitoring system using IoT technology. In *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)* (pp. 1-6). IEEE. doi.org/10.1109/OTCON56053.2023.10113976
18. Meuser, T., Zhang, W. & Hussein, A. (2024) Revisiting Edge AI: Opportunities and Challenges. University of Oulu Technical Report. https://oulurepo.oulu.fi/bitstream/10024/53248/1/nbnfioulu-202412137278.pdf

19. Miri, I., Fotouhi, A., Ewin, N.: Electric vehicle energy consumption modelling and estimation—A case study. *Int. J. Energy Research* 45(1), 501–520 (2021). https://doi.org/10.1002/er.5700
20. Mivehiclei, S., Foti, S., Testa, A., De Vehicleo, S., Sergi, F., Andaloro, L., Aloisio, D., Leonardi, S.G., Napoli, G.: Effect of WLTP CLASS 3B Driving Cycle on Lithium-Ion Battery for Electric Vehicles. *Energies* 15(18), 6703 (2022). https://doi.org/10. 3390/en15186703
21. Morali, U.: Influence of convective heat transfer coefficients on thermal behaviour of a lithium-ion cell: a numerical study. *Turkish Journal of Chemistry* 48(1), 128–136 (2024). https://doi.org/10.55730/1300-0527.3645
22. Narayanan, A., De Sena, A.S., Gutierrez-Rojas, D., Carrillo Melgarejo, D., Hussain, H.M. and Ullah, M., 2020. Key advances in pervasive edge computing for industrial Internet of Things in 5G and beyond. *IEEE Access*, 8, pp.206734–206754. https://doi.org/10.1109/ACCESS.2020.3037717
23. Nițulescu, I.-V., Korodi, A.: Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions. *IoT* 1(1), 76–91 (2020). https://doi.org/10.3390/iot1010005
24. Pamadi, V.N. and Singh, P., 2025. Edge AI vs Cloud AI: A Comparative Study of Performance, Latency and Scalability. *International Journal of Research in Modern Engineering and Emerging Technology*, 13(3). Available at: https://ijrmeet.org/edge-ai-vs-cloud-ai-a-comparative-study-of-performance-latency-and-scalability/
25. Piggott, A.J., Allen, J.S. & Pesaran, A.A., 2025. High-heat transfer lithium-ion batteries: A new era in battery thermal management. *Applied Thermal Engineering*, 272, 126347. doi.org/10.1016/j.applthermaleng.2025.126347
26. Punitha, S., Majumdar, A., Revanth, M.: TinyML-Based Battery Health Estimation for Electric Vehicles Using Kronecker-Compressed LSTM on Embedded BLE Microcontroller. *J. Electrical Systems*, 20-11s, 4947-4958, 2024. https://doi.org/10.52783/jes.8868
27. Singh, R. and Gill, S.S., 2023. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 3, pp. 71–92. Available at: https://doi.org/10.1016/j.iotcps.2023.02.004

28. Soy, H. Edge AI-Assisted IoV Application for Aggressive Driver Monitoring: A Case Study on Public Transport Buses. *International Journal of Automotive Science and Technology*, 7(3), 213-222, 2023. https://doi.org/10.30939/ijastech. 1335390
29. Sun, D., Lan, F., Zhou, Y., Chen, J.: Control algorithm of electric vehicle in coasting mode based on driving feeling. *Chinese Journal of Mechanical Engineering* 28(3), 479–485 (2015). https://doi.org/10.3901/CJME.2015.0317.032
30. Szumska, E.M. and Jurecki, R., 2022. The analysis of energy recovered during the braking of an electric vehicle in different driving conditions. *Energies*, 15(15), p.9369. Available at: doi.org/10.3390/en15249369
31. Taylor, N. Specific Heat Capacity of Lithium Ion Cells (2022). Available: https: //www.batterydesign.net/specific-heat-capacity-of-lithium-ion-cells/
32. United Nations Economic Commission for Europe (UNECE): UN Regulation No. 154 — Worldwide harmonized Light vehicles Test Procedure (WLTP). Technical report, UNECE, Geneva (2021). https://unece.org/sites/default/files/2021-08/ R154e.pdf
33. Valladolid, J., Calle, M. and Guiracocha, A., 2023. Analysis of regenerative braking efficiency in an electric vehicle through experimental tests. *Ingenius, Revista de Ciencia y Tecnología*, (29), pp.24–31. doi.org/10.17163/ings.n29.2023.02
34. Wang, T., Liu, X., Qin, D., Duan, Y.: Thermal Modeling and Prediction of the Lithium-ion Battery Based on Driving Behavior. *Energies* 15(23), 9088 (2022). https: //doi.org/10.3390/en15239088
35. Xu, J., Tu, R., Ahmed, U., Amirjamshidi, G., Hatzopoulou, M., Roorda, M.J.: An eco-score system incorporating driving behavior, vehicle characteristics, and traffic conditions. *Transp. Res. Part D: Transp. Environ.* 95, 102866 (2021). https://doi.org/10.1016/j.trd.2021.102866
36. Xu, N., Li, X., Liu, Q., Zhao, D.: An Overview of Eco-Driving Theory, Capability Evaluation, and Training Applications. *Sensors* 21(19), 6547 (2021). https://doi. org/10.3390/s21196547
37. Zhang, L.W., Yu, Y.S.: Simulation of thermal characteristics of lithium batteries for electric vehicles. *IOP Conference Series: Materials Science and Engineering* 657, 012036 (2019). https://doi.org/10.1088/1757-899X/657/1/012036

38. Zhang, Y., Chen, L. & Wu, H. (2024) Tiny Deep Ensemble: Uncertainty Estimation in Edge AI Accelerators. arXiv preprint, arXiv:2405.05286. Available at: https://arxiv.org/html/2405.05286v1
39. Zhu, Y.: Vehicular Edge Computing: A Survey on Architecture and Task Offloading. *Vehicular Edge Computing*, (2023).

Appendix 1: Data Ingestion & Parsing

```
if (!LittleFS.begin(true))
inFile = LittleFS.open("/WLTC_Class3B_1Hz_full.csv", FILE_READ);
if (inFile) inFile.readStringUntil('\n');

float val[11] = {0}; int idx = 0;
while (idx < 11 && row.length())  { /* comma-split to val[] */ }
```

Appendix 2: KPIs & Formulas

```
totalDist += speed * (dt / 3600.0);
if (coasting) coasting_seconds++;
if (current < 0) charge_sum += fabs(current); else discharge_sum += current;
energy_sum += energy_inst; regen_sum += e_recovered; ke_lost_sum += ke_lost;

float batt_eff      = (charge_sum+discharge_sum>0) ?
(charge_sum/(charge_sum+discharge_sum))*100 : 0;
float avgWhPerKm    = (totalDist>0.01) ? energy_sum/totalDist : 0;
float regen_eff_avg = (ke_lost_sum>0) ? (regen_sum/ke_lost_sum)*100 : 0;
float coasting_percent = (sec>0) ? (coasting_seconds/(float)sec)*100 : 0;
```

Appendix 3: Scoring & Status Mapping

```
energy_score  = (avgWhPerKm < 140) ? 100 : (avgWhPerKm < 180 ? 70 : 40);

regen_score   = (regen_eff_avg >= 70) ? 100 : (regen_eff_avg >= 40 ? 70 : 40);

temp_score    = (battery_temp >= 25 && battery_temp <= 40) ? 100 : 60;

current_score = (batt_eff > 50) ? 100 : (batt_eff >= 20 ? 70 : 40);

coasting_score= (coasting_percent < 5) ? 40 : (coasting_percent <= 12 ? 70 : 100);

float driver_score = energy_score*0.35 + regen_score*0.25 + temp_score*0.15
                   + current_score*0.15 + coasting_score*0.10;


String coasting_status = (coasting_percent < 5) ? "Low Coasting"
                       : (coasting_percent <= 12) ? "Moderate Coasting" : "High Coasting";

energy_status  = (avgWhPerKm < 140) ? "Efficient" : (avgWhPerKm < 180) ? "Moderate"
: "Inefficient";

current_status = (batt_eff  > 50)  ? "High" : (batt_eff >= 20) ? "Moderate" : "Low";
```

Appendix 4: SOC/Range Estimation Assumptions

```
float batt_capacity_kWh = 52; // Renault Zoe R135 (2024)

float soc_percent = ((batt_capacity_kWh*1000.0 - energy_sum) /
(batt_capacity_kWh*1000.0)) * 100.0;

float range_left_km = ((soc_percent / 100.0) * batt_capacity_kWh * 1000.0) /
emaWhPerKm;
```

Appendix 5: Edge-AI Inference

```
features[0]=speed; features[1]=distRow; /* ... */ features[9]=battery_temp;

signal_t signal; numpy::signal_from_buffer(features,
EI_CLASSIFIER_NN_INPUT_FRAME_SIZE, &signal);

run_classifier(&signal, &result, false); // temp_rise_pred set
```

Appendix 6: Telemetry Schema

```
String json = "{"
  "\"time\":\""+timeLabel+"\","
  "\"speed\":"+String(speed,1)+","
  /* ... all fields ... */
  "\"driver_score\":"+String(driver_score,1)+
"}";
client.publish(mqtt_topic, json.c_str());
```

Appendix 7: Connectivity Helpers

```
bool connectMQTT(){ String id="ESP32Client-"+String(random(0xffff),HEX);
return client.connect(id.c_str()); }
```