Sustainable Energy Management System for AIoT Solutions Using Multivariate and Multi-step Battery State of Charge Forecasting

Farnaz Kashefinishabouri

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Farnaz Kashefinishabouri**

Entitled: **Sustainable Energy Management System for AIoT Solutions Using Multivariate and Multi-step Battery State of Charge Forecasting**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________ Chair
Dr. Jeremy Clark

______________________ Examiner
Dr. Chun Wang

______________________ Supervisor
Dr. Nizar Bouguila

______________________ Co-supervisor
Dr. Zachary Patterson

Approved by ______________________
Dr. Chun Wang, Graduate Program Director
Department of Concordia Institute for Information Systems Engineering

December 2023 ______________________
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Sustainable Energy Management System for AIoT Solutions Using Multivariate and Multi-step Battery State of Charge Forecasting

Farnaz Kashefinishabouri

The convergence of Artificial Intelligence (AI) with Internet of Things (IoT) technologies, often referred to as AIoT, is transforming aspects of modern life, such as smart cities. This transformation, however, brings with it challenges, including energy management. In addressing this issue while upholding responsible AI principles, it is important to prioritize the sustainability of AIoT solutions by a promising approach which is using renewable energy sources. While renewable energy offers numerous advantages, its intermittent nature necessitates effective power management systems. Developing a power management system serving as a decision-making platform for AIoT-driven solutions is the goal of this study. This platform contains two critical components: accurate forecasts of battery "State of Charge" (SoC), and the implementation of appropriate control strategies. These strategies include adjusting energy consumption patterns to ensure stable and reliable system operation. This study focuses on accurate battery SoC forecasting, to this end, an experiment has been designed, and a data logging system has been developed to produce suitable data since publicly available datasets do not align with the specific characteristics and requirements of the research. The SoC forecasting in this study has been addressed as a multivariate and multi-step time series forecasting problem, where various machine learning and deep learning models including Decision Tree (DT), Random Forest (RF), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), and Bidirectional Gated Recurrent Unit (Bi-GRU) were benchmarked. Extensive evaluations have been conducted for different forecasting horizons on datasets with varying time intervals. It is concluded that the Bi-GRU model outperformed other models across datasets with varying time intervals and forecast horizons according to Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) evaluation metrics.

Acknowledgments

Foremost, I would like to extend my heartfelt gratitude to my supervisors, Professor Nizar Bouguila and Professor Zachary Patterson, for their unwavering support, guidance, and mentorship throughout my academic journey. This work would not be possible without their advice and support.

I am deeply grateful for the support and collaboration provided by BusPas Inc. The opportunity to work with an innovative organization has enriched my academic experience and broadened my perspective on real-world applications of the research.

I am appreciative of the support and valuable experience I gained through the Mitacs Acceleration Program.

Additionally, I would like to express my sincere thanks to my friends and all those who have helped me throughout this journey.

Finally, I would like to thank my beloved parents, and my brother for all their support throughout my life.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
IoT	Internet of Things
AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
SCiNe	Smart City Network
SoC	State of Charge
STC	Standard Test Conditions
SCC	Solar Charge Controller
MPPT	Maximum Power Point Tracker
DOE	Design of Experiments
DT	Decision Tree
RF	Random Forests
ET	Extra Trees
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
RNNs	Recurrent neural networks
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short Term Memory
Bi-GRU	Bidirectional Gated Recurrent Unit

Chapter 1

Introduction

1.1 Problem Statement

In our increasingly interconnected world, the Internet of Things (IoT) has emerged as a transformative technology, changing how data is collected and used across various domains. As a broad definition, IoT refers to the interconnected networking of everyday objects, often embedded with intelligent capabilities [1]. As a result of this widespread adoption of IoT technology, many remarkable innovations have taken place, particularly in the area of smart cities. These innovations are, however, accompanied by significant challenges. The extensive volume of data collected from different sensors and devices resulting from IoT solutions poses privacy issues that affect their widespread adoption, as this data often contains sensitive information [2]. The transmission of large amounts of data from IoT devices requires large bandwidth, and not all environments support it [3]. Additionally, IoT data is often unstructured, large-scale and unclean, which poses challenges in extracting meaningful insights from it [4]. On the other hand, a valuable contribution of Artificial intelligence (AI) is its capability to uncover insights through data analysis. When speaking of AI, it is imperative to emphasize the importance of Responsible AI, which takes into account privacy, ethics, and sustainability. Responsible AI ensures that AI technologies are developed and used in ways that protect individual privacy, maintain ethical standards, and reduce environmental impact by promoting energy-efficient practices and energy management [5]. AI integration with IoT ecosystems has empowered them with the capability to analyze vast amounts of data, derive insights, and facilitate autonomous decision-making. The convergence of AI and IoT technologies, known as AIoT, is introducing a new era of smart and adaptive systems, accelerating innovation and increasing efficiency across a variety of industries [6]. Smart cities are one of the sectors where AIoT is bringing significant innovation. In order to leverage the full power of AIoT, it is important to address IoT challenges while adhering to responsible AI principles, in which privacy, ethics, and sustainability are all taken into account. As a result, BusPas Inc., a Montreal-based company, is offering a smart platform using AIoT in order to tackle these challenges and meet the current demand for smart mobility. Their concept involves replacing regular bus stop signs with an intelligent and interconnected display named "SCiNe" which stands for "**S**mart **Ci**ty **Ne**twork". Besides showing real-time information regarding bus schedules, this device is equipped with several IoT sensors, including a light sensor, various environmental sensors, an infrared sensor, microphones, speakers, and a fisheye camera. These sensors enable the intelligent display to collect valuable data at bus stops, making them strategic hubs for gathering information on passenger and vehicle flows, environmental conditions, traffic patterns, ridership, and more. This data contributes to enhancing mobility and optimizing transportation services. Despite this, what truly distinguishes this intelligent display is its embedded computational capabilities. Through the use of AI on the edge, this device enables real-time data processing at bus stops while preserving user privacy by only transmitting descriptive information, not only safeguarding privacy but also streamlining data transmission by reducing the volume of data that needs to be transmitted after processing.

The advancement of AI technologies has resulted in larger and more powerful models capable of performing complex tasks. However, these larger models are energy-intensive, which poses a challenge. Furthermore, IoT technologies are developing rapidly, creating new challenges, one of which is managing energy resources efficiently [7]. This challenge is further compounded when integrating AI with IoT - in AIoT-driven solutions. To address these challenges and uphold the principles of responsible AI, it is important to ensure the sustainability of these solutions. The use of renewable energy sources is a promising approach to addressing this challenge. Following this approach, SCiNe is powered by a lithium-ion battery that derives its charge from a solar panel, showcasing its dedication to sustainability.

Energy generated from renewable sources provides numerous benefits, including reducing reliance on finite fossil fuels and the potential cost reductions [8, 9, 10]. Additionally, renewable energy sources such as solar panels can be harnessed in various locations, reducing reliance on centralized power grids and increasing energy self-sufficiency. However, there are several challenges associated with renewable energy that need to be addressed for it to be successfully used to power AIoT devices. Due to the intermittent nature of these energy sources, such as solar and wind power, they need to be carefully managed to ensure a consistent energy supply [11]. Factors such as weather patterns and seasonal variations can make predicting renewable energy production difficult [12]. Moreover, when renewable energy sources do not produce enough power, energy storage is required to ensure a reliable energy supply. However, current storage technologies have capacity and efficiency limitations [13]. Therefore, an effective power management system is essential to mitigate these challenges. In systems based on renewable energy sources, predictive control techniques have recently been offered as a way to cope with uncertainty and intermittency of energy production and consumption [14]. The State-of-Charge (SoC) of batteries, which indicates the amount of energy stored in them, is one of the main parameters for the development of these techniques. Consequently, effective power management of renewable energy systems involves two key aspects: accurately forecasting battery SoC, and the implementation of appropriate control strategies such as adjusting energy consumption patterns, to ensure stable and reliable system operation. This power management system serves as a decision-making system, defining different service levels for the device. At each service level, certain functionalities of the device are limited to control power consumption. Based on predicted SoC of a battery, and considering weather and solar conditions in the future, the system switches between these service levels dynamically. The main objective of this study is to develop an effective power management system for AIoT solutions, exemplified by our work on SCiNe, with a specific focus on accurate battery SoC forecasting. This SoC forecasting plays a central role within the broader decision-making system, all the while making efficient use of renewable energy to ensure the practical sustainability of AIoT solutions.

1.2 Contributions

The majority of previous research in battery State-of-Charge (SoC) forecasting has concentrated on microgrid systems, electric vehicles, and grid ancillary services. However, this study distinguishes itself by focusing on a unique application domain, which involves developing a power management system for a battery-powered AIoT device charged through a solar panel. This application introduces unique challenges and features that differentiate it from traditional scenarios in SoC forecasting. Notably, publicly available datasets, which have often been the foundation for previous research, are not directly applicable to our specialized problem domain. The contributions of this study can be summarized as follows:

1. **Designing a Unique Experimental Setup:** To address the scarcity of publicly available datasets suitable for our specific problem domain of battery SoC forecasting, we designed and implemented a unique experimental setup. To meet our research objectives, this setup was carefully designed to collect data that is directly relevant to our research.

2. **Development of a Custom Data Logging System:** To facilitate our experimentation, we configured and programmed a custom data logging system. This system played a pivotal role in gathering relevant and accurate data, ensuring the availability of appropriate data for subsequent analysis and model development.

3. **Development of a Battery SoC Estimation Tool:** Unlike other battery parameters, such as voltage, current, and temperature, State-of-Charge (SoC) cannot be directly measured. In this contribution, we employed a data-driven method to develop a battery SoC estimation tool for the SCiNe device. This tool provides a means to measure the battery's SoC, addressing the challenge where direct measurement is not possible.

4. **Multivariate and Multi-Step Time Series Forecasting for Battery SoC Forecasting:** This study addresses the challenge of Forecasting battery SoC as a multivariate and multi-step time series, employing a diverse range of forecasting models, including machine learning and deep learning approaches. We conducted a comprehensive evaluation of the forecasting models, providing a benchmark for their performance.

1.3 Thesis Overview

The rest of this thesis is organized as follows:

- In Chapter 2, we provide a literature review, exploring the state of the art in studies related to battery State-of-Charge (SoC) forecasting across different domains. Additionally, this chapter comprehensively covers the forecasting models that have been employed for battery SoC forecasting.

- In Chapter 3, we lay the foundation for our research by detailing the experimental design, setup and data collection. An overview of the plan for the experiment we designed, as well as the configuration of the data logging system, is presented in this chapter. We describe the methodologies employed to ensure the collection of relevant and precise data, which forms the basis of our subsequent analysis and model development.

- In Chapter 4, we employed a data-driven method to develop a battery SoC estimation tool for the SCiNe device, enabling direct estimation of the battery's SoC. Then, as part of the power management system, we preprocessed the collected data and applied a variety of multivariate and multi-step time series forecasting models on the data for forecasting battery SoC for different forecasting horizons. Subsequently, as part of our evaluation process, we provide a comprehensive benchmark to assess the efficiency and accuracy of these models. In conclusion, we provide important insights and future research directions, demonstrating the significance of our research.

Chapter 2

Literature Review

Energy storage becomes necessary as a consequence of using renewable energies when they fail to generate sufficient power. Batteries are commonly used for energy storage in Renewable Energy Source (RES) systems and Lithium-ion batteries are often used in this context. The SoC of batteries is one of the main parameters to be used in predictive control algorithms in the power management of systems using renewable energies. The process of determining the current state of charge of a battery is known as SoC estimation, whereas SoC forecasting involves predicting the future state of charge based on past data and other factors. While numerous methods exist for SoC estimation in batteries, limited research has been conducted specifically on SoC forecasting which is an important aspect of battery management systems (BMS) reliant on renewable energy sources, as it enables the development of effective power management strategies [15]. Unlike other battery parameters, such as voltage, current, and temperature, SoC cannot be directly measured. Researchers have studied various methods for SoC estimation, which can be divided into three categories: conventional, model-based, and data-driven methods [16]. Conventional techniques are easy to understand and effective in terms of computation, such as the ampere-hour integral or Coulomb counting methods, the open-circuit voltage (OCV) based method, and impedance tracking. However, they can suffer from accuracy degradation due to error accumulation and parameter variation caused by temperature and aging. Model-based methods use equivalent circuit models (ECMs) for batteries for SoC estimation. The two main ECMs are electrochemical ECM, which models electrochemical reactions but requires complex equations and lacks scalability for large-scale systems [17], and electrical ECM, represented by a simple circuit with a voltage source, resistance, and capacitor. When compared with other approaches, model-based methods, which incorporate adaptive filter algorithms like Kalman filters, generally achieve higher accuracy and minimize errors [18]. Data-driven methods for SoC estimation, often referred to as model-less approaches, do not rely on a battery model or in-depth knowledge of the battery. Numerous data-driven techniques have been investigated, including tree-based models like Decision Tree (DT) [19] and Random Forest (RF) [20], Neural Network (NN) [21], Fuzzy Logic [22], Support Vector Machine (SVM) [23], and Genetic Algorithm (GA) models [24]. By training these models with datasets containing SoC-related parameters such as voltage, current, and temperature, they can achieve highly accurate SoC estimation. However, the limitation of such methods is that they generally require a large amount of training data and perform best under similar conditions to those in which they were trained [25]. The main focus of this study is on SoC forecasting rather than estimation. Several studies have investigated SoC forecasting across different domains, employing various models and techniques. Researchers have used several time series methods to forecast battery SoC in the field of grid ancillary services. For instance, in [26] Ardiansyah et al. developed an approach for multi-step SoC forecasting of battery energy storage systems (BESS) in grid ancillary services, using Long Short-Term Memory (LSTM) neural networks. This model considers dynamic grid conditions and varying power demand, enabling accurate and reliable predictions of the battery SoC over multiple time steps. [27] proposed a Seq2Seq regression approach for multivariate and multi-step forecasting of BESS in frequency regulation service. The model showed promising results in accurately predicting SoC, enabling efficient use of BESS. Another domain in which SoC forecasting has been studied, is in Micro grids, such as [15]. That paper developed an embedded system for real-time forecasting of battery SoC, enabling effective energy management and decision-making in microgrid systems. In [14], MAPCAST, an adaptive control approach enhanced by predictive analytics, one of which is SoC forecasting, is used to achieve energy balance in microgrid systems. By combining predictive analytics with adaptive control techniques, the proposed method optimizes energy usage and ensures a stable energy balance, leading to improved energy management and reliable operation in microgrid systems. One of the key concerns of electric vehicle (EV) customers is the driving range which depends mainly on battery capacity. Forecasting battery SoC is therefore useful in this context as well. In [28] NaitMalek et al. introduced a hybrid method for accurate SoC forecasting in EVs. The approach combines a machine learning algorithm with an EV model to forecast battery SoC. The machine learning algorithm predicts vehicle speed, which is then used as input for the EV model to determine the battery SoC. The work presented in [29] also explored the integration of predictive analytics techniques for multi-horizon forecasting of battery SoC and contributes to the development of an intelligent management system for battery-powered electric vehicle. A variety of forecasting techniques and algorithms have been applied in battery SoC forecasting. [28] used linear regression for SoC forecasting, which offered simplicity and interpretability, making it suitable for real-time SoC forecasting in battery-powered electric vehicles. However, to address the limitation of not capturing complex nonlinear patterns in linear regression, alternative algorithms such as decision trees (DT) were employed in [30]. It is worth noting that decision trees can be prone to overfitting, and ensemble methods like random forests (RF) or gradient boosting can be employed to mitigate this issue and further enhance SoC forecasting performance. In light of this, [30] also applied random forest and Light Gradient Boosting Machine (LightGBM), whereas [29] leveraged the power of Extreme Gradient Boosting (XGBoost). Moreover, in [26], advanced deep learning architectures including LSTM, Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory (Bi-LSTM), and Bidirectional Gated Recurrent Unit (Bi-GRU) were investigated. These recurrent neural network variants demonstrated their ability to capture temporal dependencies and long-term patterns in SoC data, enabling more accurate multi-step forecasting of battery SoC. [27] proposed a solution to the challenges of multi-step SoC forecasting by using a sequence-to-sequence (seq2seq) model in deep regression learning. This model has demonstrated its effectiveness and robustness in various scenarios, making it a reliable approach for accurate multivariate and multi-step forecasting, as supported by previous studies [31]. In summary, prior research on battery SoC forecasting has been mostly centered around domains such as microgrid systems, electric vehicles, and grid ancillary services. The research presented here focuses on battery SoC forecasting that serves as the foundation for the intelligent power management system designed to optimize the operation of a battery-powered AIoT device using solar panel energy, addressing a crucial gap in the field of SoC forecasting - specifically for this unique application. While existing datasets have contributed significantly to the field of battery SoC forecasting, it is crucial to note that the majority of these datasets are tailored to specific applications, such as electric vehicles (EVs) or grid ancillary services. Upon careful examination, it became evident that the features included in these datasets were primarily focused on the characteristics of the specific application, such as vehicle speed in the case of EVs. Given the distinctive nature of our study, which revolves around the intelligent power management system for a battery-powered AIoT device utilizing solar panel energy, the available datasets did not align with the essential features required for our analysis. Considering the specific characteristics and scale of this study, a custom dataset was required and created since publicly available datasets did not meet the study's needs and lacked the necessary features. As such, in this thesis, firstly, in order to overcome the shortage of suitable datasets in this domain, an experimental setup was designed and a custom data logging system was devised to ensure the acquisition of relevant and accurate data. Following the crucial step of data collection, this study used various forecasting models, including machine learning and deep learning approaches, to forecast battery SoC as a multivariate and multi-step time series problem. Lastly, it conducted a comprehensive evaluation of the forecasting models, providing valuable insights into their performance and their usage in the domain of power management for autonomous solar-powered AIoT devices.

Chapter 3

Experimental Framework: Design, Setup, and Data Collection

In the pursuit of developing an effective power management system for AIoT solutions, this study works on developing a tailored system, with the SCiNe serving as an example of AIoT technology. As a first step toward achieving this goal, it is essential to gain a comprehensive understanding of the SCiNe, including its various subsystems, their functions, and power consumption associated with each. This foundational knowledge of the SCiNe is pivotal in the design of the experiment and the development of the data logging system to gather the requisite data for this study. In the following sections of this chapter, a detailed overview of the SCiNe device is provided. The exploration begins with an examination of the different subsystems within the SCiNe, highlighting their individual functions and power consumption. Subsequently, the chapter delves into the experimental design, tailored for capturing the data required for the power management system. Finally, it discusses the configuration and programming of the data logging system that played a key role in data collection and provides an overview of the data gathered from the experiments.

3.1 SCiNe

The development of the SCiNe is part of the concept of smart mobility and smart cities. Using the power of the Internet of Things (IoT), the SCiNe contributes to the development of mobility as a service (MaaS) [1], promotes multimodality from bus stops and allows the integration of additional applications in an ecosystem dedicated to the city. The SCiNe is designed to be a building block for smart mobility. Deploying the SCiNe displays at bus stops and enabling them to interact with their environment enhances the transport service, line efficiency, service quality, and customer experience. This technology connects the transit agency to its passengers at the bus stop, aiming to increase ridership and develop a lasting relationship of trust with users. Figure 3.1 provides a visual representation of the SCiNe device.

Figure 3.1: SCiNe (Smart City Network)

3.1.1 SCiNe Subsystems

The SCiNes consist of many interconnected subsystems, each of which has specific functions that contribute to its overall capabilities. The display subsystem of the SCiNe currently serves as an information hub, providing real-time bus schedules and essential travel details to passengers. Besides the display, the speaker subsystem offers an auditory interface, allowing passengers to activate

[1]MaaS refers to the integration and access to various transportation services (such as public transportation, ride-sharing, car-sharing, bike-sharing, scooter-sharing, taxi, automobile rental, ride-hailing, and so on) in a single digital mobility offer [32].

and receive bus schedule announcements by pressing a button integrated into the SCiNe pole. This feature, useful for individuals with disabilities, enhances accessibility and ensures that everyone can access information. Additionally, the SCiNe is equipped with a camera and computational source, enabling advanced computer vision capabilities. This subsystem is currently operational, playing a crucial role in real-time passenger counting at bus stops. The passenger counting feature provides valuable data to transit agencies for informed decision-making in transportation planning and scheduling adjustments. Notably, the SCiNe processes this data in real-time on the edge, ensuring passenger privacy by transmitting only descriptive data, such as passenger counts, without compromising personal information. In addition to these features, the SCiNe employs a combination of motion and light sensors along with LEDs to enhance the safety and user experience at bus stops, particularly during low-light conditions or darkness. The motion sensor is designed to detect the presence of passengers or pedestrians near the bus stop. Simultaneously, the light sensor measures the ambient brightness of the bus stop environment. When someone approaches the bus stop and the light sensor registers that the illumination falls below a predefined threshold, a security LED and display LED are activated. These LEDs provide valuable visibility and security for passengers during nighttime or poorly lit conditions. This feature ensures that waiting passengers can navigate the bus stop area more safely and comfortably. In cold weather, the SCiNe activates heaters to maintain optimal performance of the display and battery, ensuring uninterrupted service for passengers. These heating elements ensure the SCiNe's reliability even in extremely cold conditions.

While the current SCiNe enhances passenger experience with its functioning components, there are some aspirational features that are under development as follows: The display subsystem, beyond schedule updates, is envisioned to offer dynamic content capabilities. This includes sharing a wide range of information by allowing transit agencies to display advertisements, weather forecasts, promotional content, and announcements. The microphone subsystem, while not currently implemented, is intended to provide passengers the ability to interact with the SCiNe through voice commands. Moreover, the presence of speakers and microphones lays the foundation for the development of an offline AI system, enabling advanced, AI-driven communication with passengers. The camera's capabilities, although currently used for passenger counting, are aspired to extend to detecting various issues at bus stops, such as trash buildup, contributing to a cleaner and more pleasant environment for passengers. Additionally, this feature aims to support cost-efficiency measures by identifying bus stops with low or zero passenger activity during specific time frames, potentially leading to the rerouting of transport services through the SCiNe platform, such as taxis or ridesharing, when buses are not needed. Table 3.1 provides an overview of these subsystems along with brief descriptions of their functionalities.

Table 3.1: SCiNe Subsystems and their functionalities

Subsystem	Usage Description
Display	Showing real-time bus schedules and information for passengers
Speaker	Announcements for passengers
Microphone(s)	Voice commands from passengers
Camera + lens	Passenger counter; infrastructure inspection
Motion sensor	Determine if there are passengers / pedestrians near the appliance
Light sensor	Measuring the brightness of the environment (bus stop)
LEDs	Lightening bus stop and display
Heaters	Heat display and battery during winter

Figure 3.2 illustrates how the SCiNe's subsystems are arranged and positioned in the lower section of the device, giving a visual overview of how these components are integrated.

Figure 3.2: Subsystems of the SCiNe: 1, 4: Microphones, 2: Camera and Lens, 3: Speaker, 5: Motion Sensor, 6: Security LED

The detailed view of the SCiNe's board, showing some other subsystems, can be seen in Figure 3.3.

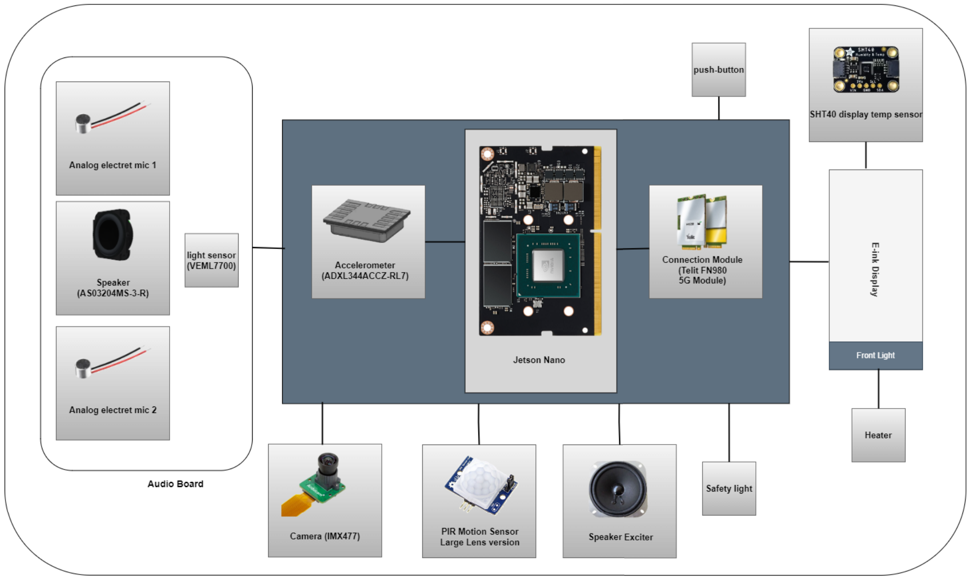

Figure 3.3: Detailed View of the SCiNe Board and other Subsystems

3.1.2 SCiNe's Power Infrastructure

SCiNes use a solar panel to recharge a lithium-ion battery and harness renewable energy sources for its operation. The power infrastructure of the SCiNe encompasses essential components: a solar panel, a lithium-ion battery, a solar charge controller, and the device itself. Each component fulfills a vital role in energy generation, storage, management, and consumption within the system. The components of the power infrastructure of SCiNe are:

- The SCiNe is equipped with a monocrystalline solar panel installed on top of its pole, designed to harness renewable energy. With a maximum power output of 50W at STC (Standard Test Conditions) [2], it provides a sustainable power source for the system's operation.
- To store the energy produced by the solar panel, the SCiNe features a lithium-ion battery with a capacity of 532WH, installed at the back of the solar panel on top of the pole. The battery's output voltage ranges from 12 to 16.8 V ensuring a steady and reliable power supply. This energy storage solution enables the SCiNe to operate even when sunlight is not available.

[2] STC stands for Standard Test Conditions in solar panel technology. It refers to a set of standard conditions under which the performance of a solar panel is measured. The STC temperature is 25°C, and the solar irradiance is $1000\,\mathrm{W/m^2}$ with an air mass of 1.5.

- At the heart of the SCiNe's power infrastructure resides the Solar Charge Controller (SCC), which harnesses MPPT (Maximum Power Point Tracker) technology to optimize the energy flow generated by the solar panel into the lithium-ion battery. This essential component continuously monitors the energy generation process, actively tracking the maximum power point by adjusting voltage and current. The MPPT controller ensures that the battery receives the highest possible charging current, enhancing energy efficiency. This device acts as a bridge linking the solar panel, battery, and the SCiNe itself, enabling seamless energy transfer while safeguarding the lithium-ion battery against overcharging or deep discharge.

- The SCiNe serves as the primary energy consumer within its power ecosystem. Comprising multiple interconnected subsystems, the SCiNe has varying power consumption requirements, each contributing to its overall functionality. Table 3.2 shows the average power consumption of these subsystems.

Table 3.2: Average Power consumptions of SCiNe Subsystems

Subsystem	Average Power Consumption (W)
Baseline [3]	3.999
Security LED	5.056
Display LED	2.870
Speaker	0.449
People counting	1.500
Heater	15.913

3.1.3 Telemetry Data

Telemetry data refers to the data collected and transferred from remote or inaccessible sources to a central point for monitoring, analysis, and control. Each SCiNe device collects Telemetry data from its various sensors to improve mobility and optimize transportation services and sends it to a cloud platform named "ORA" developed by BusPas. Thus, "ORA" acts as the central hub, to facilitate their communication and efficient management across all of these interconnected devices.

[3] The "Baseline" power consumption represents the total power consumption of all subsystems except those mentioned in the subsequent rows (i.e., LEDs, Speaker)

Some of the parameters existing in the Telemetry data related to the subsystems' power consumption are shown in Table 3.3.

Table 3.3: Telemetry Data

Telemetry Parameter	Telemetry Parameter Unit
InputVoltage	Volts
InputCurrent	Amps
Button	Boolean
Security LED Brightness	Percentage of brightness
Display LED Brightness	Percentage of brightness

In terms of data requirements for the design of the power management system for the SCiNe, it is imperative to collect data related to each component of the SCiNe's power infrastructure: the solar panel (energy generator), the lithium-ion battery (energy storage), and the SCiNe subsystems (energy consumer). While telemetry data offers valuable insights into the SCiNe subsystems, gathering data related to the battery and the solar panel is essential. These components, being external to the SCiNe board, require a distinct data collection process. This data collection can be efficiently executed through the solar charge controller, which serves as a link between the battery, the SCiNe, and the solar panel. Collecting data directly from this central component is a fundamental step in developing an effective power management system for the SCiNe. To address this requirement, an experiment has been designed to simulate a real situation and in order to conduct the experiment, an experimental setup has been configured and programmed to collect relevant and accurate data.

3.2 Design of Experiment

Design of experiments (DOE) refers to a structured and systematic approach to planning and conducting experiments to obtain meaningful and reliable results [33]. An experimental plan has been designed to observe and analyze the behavior of the SCiNe's power infrastructure. The battery charging and discharging methodology were the two crucial aspects that the experimental plan focused on. Since the experiment was conducted in a controlled lab environment without direct sunlight, a programmable power supply was used to charge the battery instead of solar panel. To simulate the charging effect of the solar panel, solar radiation data for the desired location were obtained from publicly available sources [34]. Based on solar radiation values for each hour, Equation 1 was used to calculate the voltage and current settings for the power supply simulator. The device's solar panel specifications indicated a power output of 50W under standard test conditions (STC) with solar radiation at $1000W/m^2$. This information was used to calculate the amount of power that the solar panel would deliver to the battery based on the actual solar radiation data.

$$\text{Charge Power} = \frac{\text{Solar Radiation}}{\text{STC Solar Radiation}} \times \text{Solar Panel Power at STC} \tag{1}$$

For instance, if the solar radiation for a particular hour was measured as $300\,\mathrm{W/m^2}$, the power supply's voltage and current were adjusted to deliver 15W ($\frac{300\,\mathrm{W/m^2}}{1000\,\mathrm{W/m^2}} \times 50\,\mathrm{W}$) to the battery during that hour. This approach facilitated the simulation of solar charging within the lab environment. As part of SCiNe's functionality, the device sends sensor data to the cloud at five-minute intervals which is called telemetry data. This data reflects the activation and deactivation of various subsystems of the device. In this study, for the discharging methodology, the telemetry data for the month of February 2023 was used. The collected telemetry data were employed to precisely reproduce the activation and deactivation patterns of the subsystems to simulate real-world scenarios throughout the experiment. For the discharging methodology in this experiment, we focused on activating and deactivating specific subsystems shown in Table 3.4. It's worth noting that not all the SCiNe subsystems were included in the experiment, as some are not yet fully developed, and their control options have not been implemented. To ensure consistency in the data gathering process, the experiment

Table 3.4: Considered Subsystems in the Experiment

Subsystem
Security LED
Display LED
Speaker
People Counting Process

was initiated with a fully charged battery and incorporated a repetitive cycle. When the battery SoC reached a critical level close to zero, the experiment was temporarily paused. The battery was then charged back to its full capacity, and the experiment was resumed from the point at which the battery had reached the critical level. This process of discharging, pausing, recharging, and resuming was repeated multiple times throughout the experiment, ensuring a reliable and controlled data collection approach. Through the carefully planned experiments, which encompassed the simulated solar charging and the replication of real-world subsystem activation and deactivation, a comprehensive dataset was obtained.

3.3 Experimental Setup

Experimental setup refers to the physical arrangement and configuration of the equipment, instruments, and apparatus used in an experiment. It involves the practical aspects of setting up an experiment, ensuring that the conditions are controlled, and the data can be collected accurately. To gather data for this project, a comprehensive experimental setup was designed to capture and analyze the relevant parameters of the power infrastructure of the SCiNe. This system comprised essential components, including a lithium-ion battery, a solar panel (substituted with a programmable power supply for lab environment purposes), a load representing the SCiNe with various subsystems, and an MPPT solar charge controller placed inside the battery pack. To facilitate data collection, we replaced the MPPT charge controller within the battery pack with a programmable alternative. The battery, power supply, and load were interconnected through the MPPT charge controller, while a computer served as the central control unit for data collection and control of the power supply. The computer was connected to both the MPPT charge controller and the power supply, allowing for real-time monitoring and control of the power supply's parameters. Figure 3.4 illustrates the setup configuration. Detailed photographs of the actual configuration are included in Appendix A. To control the power supply accurately based on solar data, a Python script was developed to adjust the power supply's parameters in real-time. Additionally, scripts were written to activate and deactivate the SCiNe subsystems based on telemetry data, allowing for the simulation of real-world scenarios. The comprehensive experimental setup and the development of these scripts and data logging system were pivotal to the project's success.

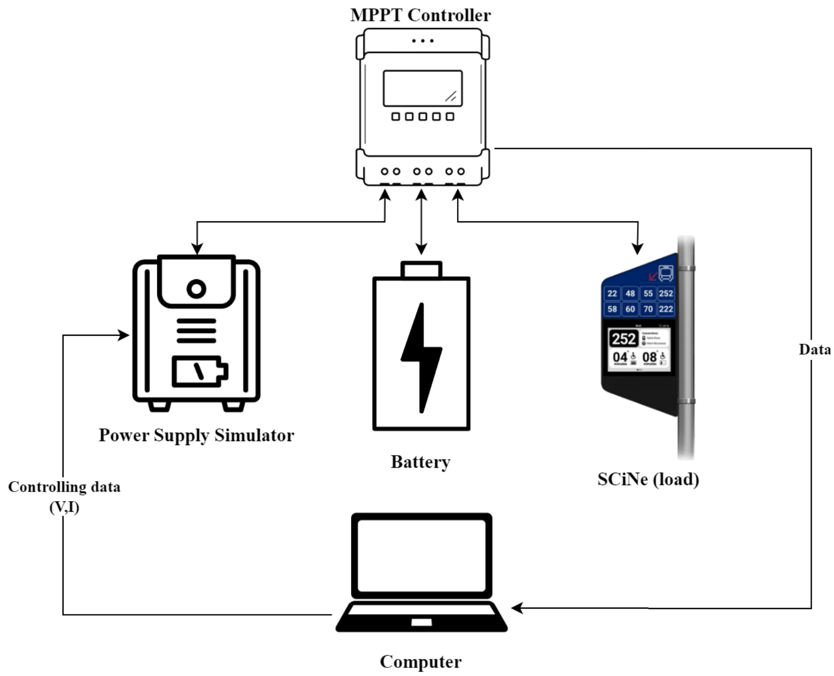

Figure 3.4: Setup Configuration

3.4 Data Collection

A data logging system was developed to collect data from the programmable MPPT charge controller at one-minute intervals. The system, developed using Python, Prometheus, and Grafana tools, ensures seamless and uninterrupted data capture. This continuous data collection process was essential to analyze and understand how the power infrastructure components interacted and performed under varying conditions. The collected parameters from the MPPT charge controller are presented in Table 3.5. The dataset produced for this study consists of 40,320 observations, which corresponds to the number of minutes in 28 days (28 days × 24 hours × 60 minutes). This dataset serves as the foundation for the subsequent analysis and model development.

Table 3.5: Description of Parameters Information Gathered from the MPPT Controller

Name	Detailed Parameters
Solar Information	Voltage, Current, Working State, Power
Battery Information	Voltage, Current, Temperature, SoC, Battery State
Load Information	Current, Voltage, Power
Controller Information	Temperature, Working State

Chapter 4

Methodology and Results

This chapter begins with the development of a specialized battery SoC estimation tool exclusively tailored to the SCiNe device, addressing the challenge of direct battery SoC estimation. It then delves into the battery SoC forecasting section, a crucial component of the SCiNe's power management system. We explore time series forecasting fundamentals, including various forecasting types, models, and the associated data preprocessing, model development, and evaluation. Additionally, this chapter introduces service levels, integral to the SCiNe's power management, and concludes by emphasizing the significance of addressing energy challenges in AIoT solutions, with a primary focus on accurate battery SoC forecasting as a foundational step toward building a sustainable power management system for AIoT applications.

4.1 Battery SoC estimation

The State of Charge (SoC) in a battery cannot be measured directly; instead, it must be estimated based on measurable parameters like current and voltage. This differs from other battery parameters, such as voltage, current, and temperature, which can be directly measured [35]. The estimation of the SoC of a battery is a fundamental aspect of managing and optimizing the operation of the SCiNe device. Having an accurate estimation of the battery SoC is crucial to ensuring the device's efficiency and reliability, since it directly impacts the availability and performance of SCiNe's services. Therefore, a data-driven approach has been adopted to develop a battery SoC estimation tool tailored to the SCiNe device. This tool leverages available data sources, namely telemetry data and solar data, to estimate the battery's SoC based on data collected from the MPPT controller. This approach offers a promising solution for addressing the challenges of estimating SoC in the context of SCiNe's power infrastructure setup. There are several stages involved in developing this tool, including data preprocessing, model development, and thorough evaluation, all of which will be elaborated on in the subsequent sections.

4.1.1 Data Preprocessing

Telemetry data and solar radiation data have been used as inputs in the development of the battery SoC estimation tool. Telemetry data provides insights into the SCiNe subsystems' behavior. In the context of power management, telemetry data features of load voltage and load current serve as crucial indicators of the system's power consumption and operation. These features, measured at 5-minute intervals, enable us to assess the power dynamics within the SCiNe device. Furthermore, the telemetry data reflects the activation and deactivation of various subsystems. The number of LEDs activations and speaker activations were considered as relevant features as these subsystems were focused on, in the experiments to collect data. Solar data, representing solar radiation, is another resource integrated into the SoC estimation model. The solar radiation data serves as a proxy for the energy input to the system, allowing us to consider the effect of solar power generation in our estimation.

Feature Correlations

Figure 4.1 displays a correlation heatmap based on Pearson correlation coefficients, illustrating the relationships between the input features and the battery State of Charge (SoC). Correlation values range from -1 to 1 in the heatmap. A value of 1 indicates a perfect positive correlation, meaning that as the input feature increases, the battery SoC also increases. Conversely, a value of -1 signifies a perfect negative correlation, suggesting that as the input feature increases, the battery SoC decreases. A correlation value of 0 indicates no linear correlation between the two variables.

The pearson correlation coefficient r is calculated using Equation 2:

$$r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}} \tag{2}$$

where S_{xy} represents the covariance between variables X and Y, S_{xx} is the variance of X and S_{yy} is the variance of Y.

Based on the heatmap, the highest correlations with battery SoC are observed for load voltage and load current, indicating their significant influence on the battery SoC. Although solar radiation, LEDs and speakers activation counts do not correlate with battery SoC as strongly as load voltage and load current, their inclusion as inputs still adds value to the SoC estimation model. These features, while individually having lower correlation, can still contribute to the model's performance when considered alongside other features, highlighting the potential synergy of various input features in improving the overall predictive capability of the model.

Data Resampling and Normalization

The telemetry data, collected at 5-minute intervals, provides valuable information about the SCiNe subsystems and their behavior. To align this data with the one-minute interval MPPT controller data, we resampled it to 5-minute intervals, ensuring that both datasets could be integrated effectively. Normalization is another step in data preprocessing. This process brings all the input features and the output variable to a common scale. By scaling the data to the same range, each feature contributes proportionally to the model training, preventing issues related to disparate data ranges. In this study, MinMax scaling was employed, which scales the data within a range of 0 to 1 as shown in Equation 3. This approach ensures that the dataset is appropriately normalized, facilitating effective model development and training.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{3}$$

where X is the original feature value, X_{min} is the minimum value of the feature, X_{max} is the maximum value of the feature, and X_{norm} is the normalized feature value.

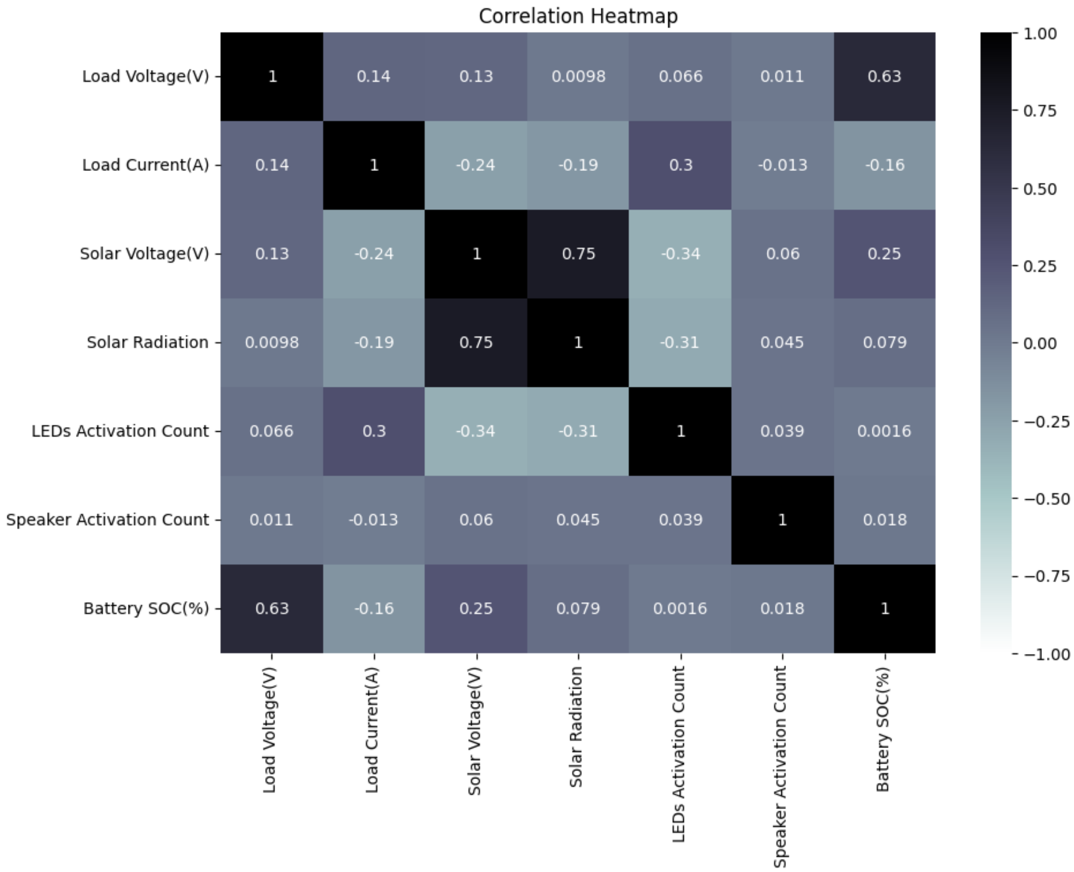

Figure 4.1: Pearson correlation heatmap displaying the relationship between the input features and the battery SoC

4.1.2 Model Development

In order to develop the battery SoC estimation model for the SCiNe device, we employed tree-based models as the primary modelling approach. We considered Decision Tree (DT), Random Forest (RF), and Extra Tree (ET) models. There are several reasons why tree-based models were chosen. The main advantage of these solutions lies in their computational efficiency, making them perfect for edge deployment. In order to maximize performance and minimize resource consumption, the SoC estimation model has to be lightweight, which can be achieved by ensuring that it runs on the SCiNe device itself. It should also be noted that battery behavior and SoC estimation often involve complex, non-linear relationships between input features and the target variable. Tree-based models are ideal for covering such non-linearities, as well as accommodating the real-world complexities of battery behavior. It is also possible to analyze feature importance with tree-based models. Using the importance scores of different input features, we can gain insight into which parameters greatly affect battery SoC, enabling us to optimize feature selection and engineering. Tree-based models also provide interpretability, which is important for understanding the model's predictions, fostering trust, and providing transparency [19].

A decision tree is a fundamental machine learning model that has been widely used across multiple applications because of its simplicity and interoperability. They are often described as hierarchical structures that recursively partition the data based on a series of binary decisions. These decisions are represented by the nodes of the tree, and the leaves contain the final predictions. Decision Trees, however, are prone to overfitting when they become too complex. They can be controlled by adjusting parameters like the maximum depth of the tree, minimum samples required to split a node, and others [36]. Based on the foundation of Decision Trees, Random Forests are a method of ensemble learning. As part of the training process, multiple Decision Trees are constructed and their predictions are aggregated to improve accuracy and reduce overfitting. The advantage of Random Forests is that they can handle large and complex data while maintaining some degree of interpretation [37]. Developed on the Random Forest approach, Extra Trees, short for Extremely Randomized Trees, are another ensemble learning method. They also create multiple Decision Trees, but with a notable difference: the Extra Tree nodes are split based on random thresholds rather than the best possible thresholds, resulting in greater diversity. As a result of this randomness, Extra Trees are less likely to overfit and are faster to train, making them suitable for situations where efficiency is crucial [38]. The dataset was divided into two subsets for the development of these tree-based models for estimating the battery SoC: a training set and a testing set. 80% of the data was allocated to the training set, while the remaining 20% was used for testing and evaluating the models' performance.

4.1.3 Evaluation

To evaluate the battery SoC estimation models, we used commonly used regression metrics, such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These metrics provide quantitative measures of the statistical quality of the estimations, allowing for a comprehensive evaluation of the accuracy in estimation of the SoC values. MAE measures the average magnitude of prediction errors. It is calculated as the mean of the absolute differences between the predicted values ($\hat{y}_i$) and the actual values (y_i) for each observation in the dataset, as shown in Equation 4. In this equation, n is the number of observations.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

RMSE is the square root of the average of the squared differences between the predicted values ($\hat{y}_i$) and the actual values (y_i) for each observation in the dataset, represented by Equation 5.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

Both MAE and RMSE express average model prediction error in units of the variable of interest and can take values from 0 to positive infinity. While MAE treats all errors equally, RMSE is more sensitive to outliers because of its squaring operation, which assigns higher weight to larger errors. Lower values for both metrics indicate better prediction performance. Table 4.1 summarizes the MAE and RMSE values for each of the tree-based models — Decision Tree (DT), Random Forests (RF), and Extra Trees (ET) — based on their performance on test data.

Table 4.1: Battery SoC estimation models' performance on test data

Algorithm	MAE	RMSE
Decision Tree	0.081	0.117
Random Forest	0.080	0.108
Extra Tree	**0.079**	**0.112**

In the evaluation of battery SoC estimation models for the SCiNe device, the performance results indicate the effectiveness of the Extra Trees model, which exhibits the best performance with the lowest MAE of 0.079 and RMSE of 0.122. As a comparison, Decision Trees and Random Forests yield similar performance, based on RMSEs and MAEs. Overall, these findings point to the efficiency of the Extra Trees model in SoC estimation, resulting in enhanced reliability and precision, which are critical for optimizing SCiNe's operation.

A comparison of actual and predicted SoC values for test data provides valuable insight into the performance of the SoC estimation models. In Figure 4.2, the scatter plot illustrates how well the predicted values align with the actual SoC values. A model's performance can be evaluated based on the proximity between data points and the 45-degree reference line, which signifies the accuracy of predictions. Extra Trees (ET) exhibit a remarkable alignment between actual and predicted values, demonstrating its effectiveness in accurately estimating battery SoC. This visualization not only confirms the quantitative performance metrics but also provides valuable insights into the distribution of errors. It further emphasizes the ET model's ability to capture complex relationships within the dataset, resulting in accurate SoC estimation.

To further enhance the models, an investigation into potential input features was conducted. The analysis revealed that "Solar Voltage", obtainable from the MPPT controller, presented a high correlation with Battery SoC as shown in Figure 4.1. The improved performance of these tree-based models after adding this feature are shown in Table 4.2.

Table 4.2: Battery SoC estimation models' improved performance on test data

Algorithm	MAE	RMSE
Decision Tree	0.040	0.079
Random Forest	0.039	0.071
Extra Tree	**0.037**	**0.064**

Incorporating this feature into the models had a significant impact on their performance. For instance, it led to a reduction in MAE to 0.037 and RMSE to 0.064 for the ET model. This reduction in error highlights the potential of incorporating additional features into the model and emphasizes its importance for accurate battery SoC estimation. In In Figure 4.3, the scatter plots confirm the improvement of the models. The predicted values align more closely with the actual values, as the points cluster tightly around the 45-degree reference line.

Feature Importance Analysis

Shapely (SHapley Additive Explanations) has emerged as a valuable tool for understanding the driving forces behind model predictions in the realm of machine learning and model interpretability. For analyzing the importance of individual features within a predictive model, Shapely offers a comprehensive framework. In order to do so, it draws inspiration from cooperative game theory and

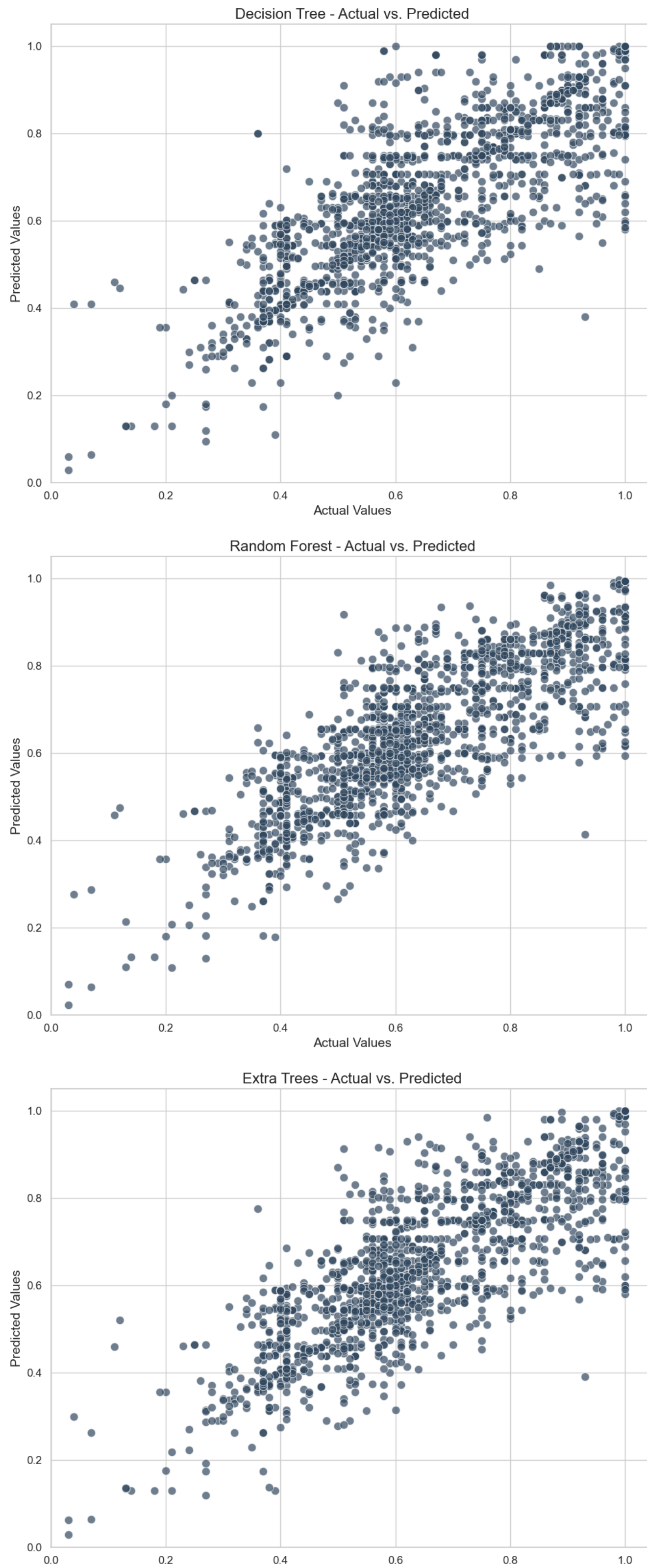

Figure 4.2: Actual vs Predicted Battery SoC Estimation Models

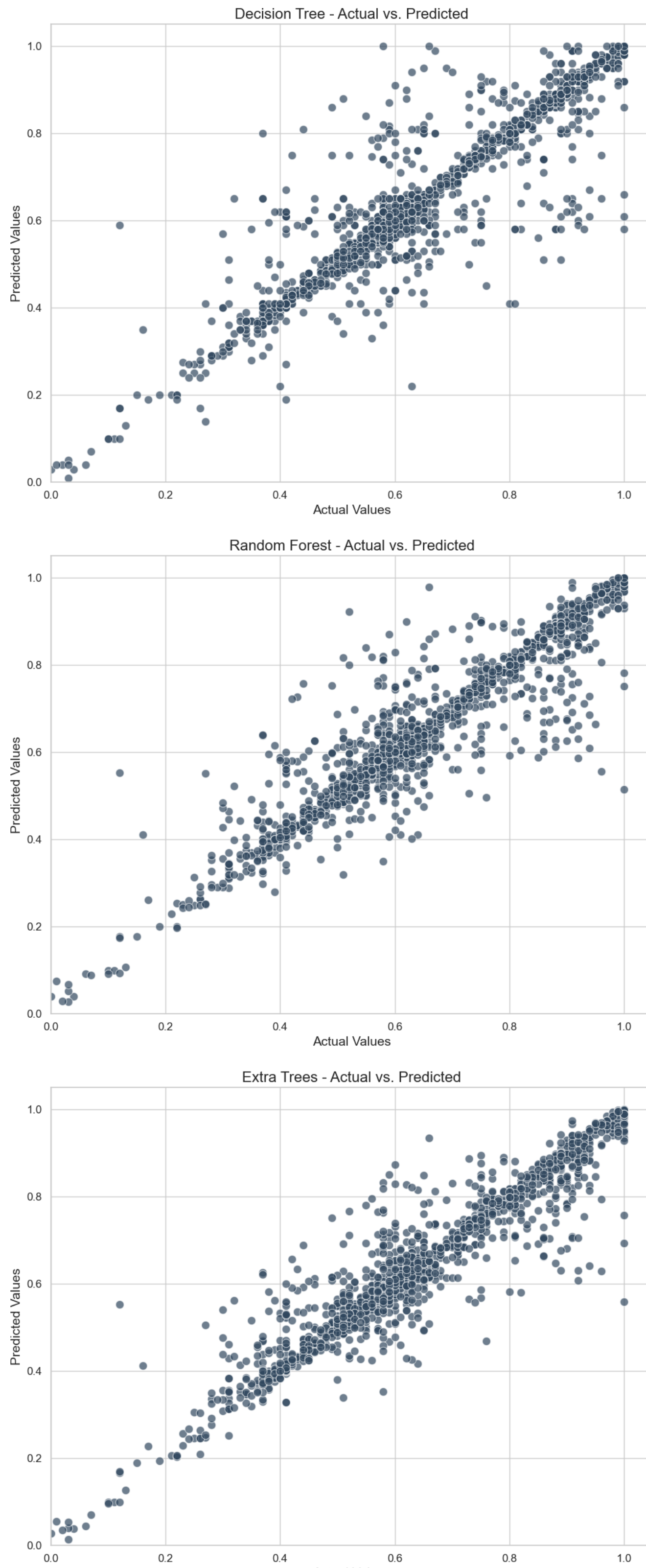

Figure 4.3: Actual vs Predicted of Improved Battery SoC Estimation Models

Shapley values, which are used to fairly distribute the contributions of the features to the prediction. Based on Shapely, each feature is given a specific “Shapley value”, which represents its impact. A Shapley value is calculated by comparing the prediction of the model with and without a feature, taking into account all possible combinations in order to calculate the importance of a feature. In practice, Shapely assigns numerical values to each feature, indicating the extent to which they influence the model’s outcomes. A deeper understanding of model behavior can be gained by exploring these Shapley values [39]. Figure 4.4 shows the SHAP Summary plot for the input features for the Battery SoC estimation models. Summary plots combine feature importance with feature effects. Each point is a Shapley value of an instance per feature. The y-axis is determined by the feature, and the x-axis is determined by the Shapley value. The features are ordered according to their importance. It is shown that “Solar Voltage” is the most important and has the highest Shapley value range among the features. This is why incorporating this feature substantially reduced the errors of the estimation models

4.2 Battery SoC Forecasting

Developing an efficient power management system for the SCiNe device relies on accurate battery State of Charge (SoC) forecasting. In renewable energy systems, managing power effectively involves two fundamental aspects. Firstly, battery SoC, a critical parameter that indicates how much energy they store, must be accurately predicted. Secondly, it necessitates the implementation of tailored control strategies, including adjustments to energy consumption patterns, all geared towards ensuring stable and reliable system operation, with these adjustments being influenced by the forecasted SoC of the battery. As part of the battery SoC forecasting approach, we explore the fundamentals of time series forecasting.

4.2.1 Time Series Forecasting

The term “time series” refers to a sequence S of historical measurements y_t of an observable variable y at equal time intervals. A time series data set typically consists of two main components:

1. Time Stamps: The time stamps represent the chronological moments when each observation

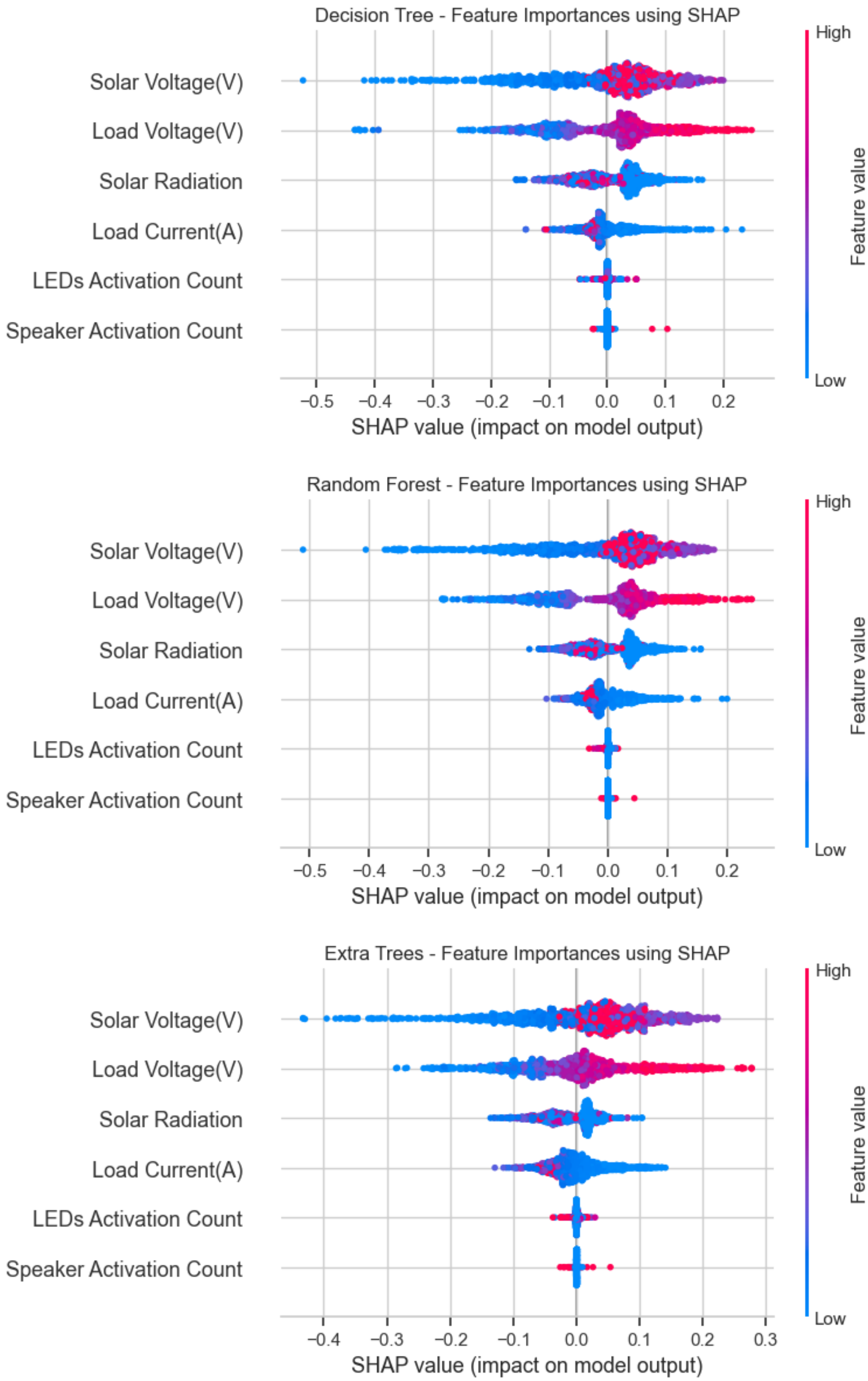

Figure 4.4: Feature Importance - SHAP values

is made. Depending on the context, time stamps can be expressed in seconds, minutes, hours, days, months, or years.

2. Values or Measurements: Data values are linked to each time stamp. In the context of battery SoC forecasting, these values indicate the state of charge of the battery at each recorded time point.

Forecasting future values of an observed time series is a fundamental practice with applications across various scientific and engineering domains [40]. In the context of this study, which centers on battery SoC forecasting, this forecasting process assumes a notably crucial role. By analyzing prior patterns and assuming that future trends will be similar to historical ones, time series forecasting enables you to forecast future events. Essentially, forecasting involves establishing models from historical data and using them to make observations and guide future strategic decisions. It involves making predictions based on historical data.

4.2.2 Types of Time Series Forecasting

A forecasting approach depends on the underlying objectives and characteristics of the data to be forecasted. Two factors determine the nature of forecasting tasks are the time horizon and the number of variables involved. Forecasting can be categorised into single-step and multi-step forecasting based on time horizon. When single-step forecasting is used, the value of the time series is predicted at a single, specific future time point. Such forecasts are often used for forecasting immediate or very short-term predictions. Multi-step forecasting consists of predicting multiple future values beyond a single time point. The goal is to forecast a sequence of future observations, typically denoted as $[y_{N+1}, \ldots, y_{N+H}]$, where $H > 1$ represents the forecasting horizon. Furthermore, univariate and multivariate forecasts differ based on its involvement of variables. Univariate Time Series Forecasting involves predicting the future values of a time series based solely on its historical data. In contrast, Multivariate Time Series Forecasting, also known as using exogenous variables, incorporates additional predictors beyond the primary time series to make forecasts [41]. Multi-step and multivariate time series forecasting techniques are used in this study in order to provide accurate forecasts.

Multi-step Forecasting Strategies

For multi-step forecasting, three strategies can be considered:

1. **Recursive Strategy:** The recursive strategy involves one-step-ahead forecasting based on actual observed values from previous steps, which serve as inputs to forecast the next step. Although adaptable, this approach can be computationally intensive.The strategy begins with the initial training of a one-step model, denoted as f, which predicts y_{t+1} using historical data $y_t, \ldots, y_{t-n+1}$ and w_{t+1}. This model is trained for t within the range of $n, \ldots, N-1$. Subsequently, the model is employed recursively to provide multi-step predictions, following the equation:

$$y_{t+1} = f(y_t, \ldots, y_{t-n+1}) + w_{t+1} \tag{6}$$

2. **Direct Strategy:** Instead of making iterative predictions one step at a time, the direct strategy efficiently generates forecasts for multiple future time steps simultaneously. This approach is computationally efficient but may require a more complex modeling approach. It involves the independent training of H models, each represented by f_h, to predict y_{t+h} based on historical data $y_t, \ldots, y_{t-n+1}$ and the term w_{t+h}. This strategy operates for t in the range of $n, \ldots, N-H$ and h in the range of $1, \ldots, H$. The multi-step forecast is derived through the following equation:

$$y_{t+h} = f_h(y_t, \ldots, y_{t-n+1}) + w_{t+h} \tag{7}$$

This equation describes the process of forecasting y_{t+h} by concatenating predictions made by the H models.

3. **Multiple Output Strategy:** The multiple output strategy revolutionizes multi-step forecasting by departing from single-output mappings. Conventional techniques model a multi-input single-output mapping, but when long-term predictions are needed in a stochastic setting, this approach may introduce biases. Instead, the multiple output strategy embraces multi-output dependencies, utilizing multi-output techniques. In this approach, the forecasted value is

represented as a vector of future values, as shown by the equation:

$$[x_{t+H}, \ldots, x_{t+1}] = F(x_t, \ldots, x_{t-n+1}) + w. \tag{8}$$

where t varies within the range from n to $N - H$. This approach accounts for complex dependencies among inputs, inputs-outputs, and among the outputs themselves, making it ideal for addressing long-term forecasting challenges [42].

The forecasting models employed in this study utilize the multi-output strategy to enhance multi-step forecasting accuracy.

4.2.3 Forecasting Models

In this study, a variety of models are employed for multi-step multivariate forecasting of Battery SoC. These models encompass both machine learning methods, such as Decision Trees (DT) and Random Forests (RF), which were previously used in the Battery SoC Estimation section, and deep learning models. The deep learning models used in this study include Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (Bi-GRU).

Long-short Term Memory

Recurrent neural networks (RNNs) are designed to effectively handle sequential data, making them suitable for applications like multi-step time series forecasting. While RNNs are capable of modeling sequences, they face a challenge known as the "vanishing gradient problem", which hinders their ability to capture long-range dependencies in data. LSTM, or Long Short-Term Memory, is an advanced RNN architecture devised to address this issue [31]. LSTM introduces key components to overcome the limitations of traditional RNNs. Its architecture includes several specialized mechanisms called "gates". These gates allow LSTM to control the flow of information over extended sequences by selectively remembering and forgetting information, thus enabling more effective learning from data. In essence, LSTM gates serve as control units that regulate the network's memory state and decision-making processes, ensuring that relevant information is retained and unnecessary information is discarded [43]. To delve deeper into the LSTM architecture, it's essential to understand the functionality of its gates. An LSTM cell architecture is illustrated in Figure 4.5, highlighting its key components and their interconnections.

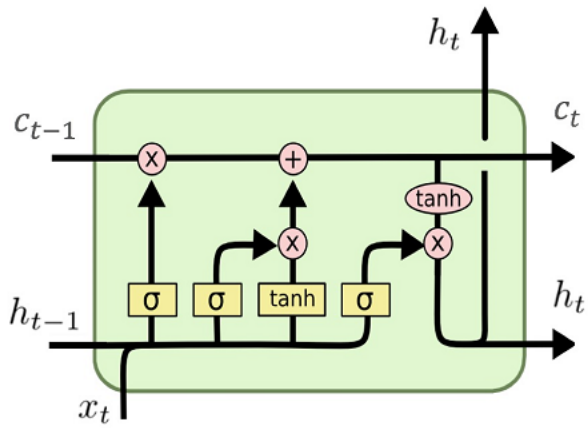

Figure 4.5: LSTM Structure [44].

In LSTM cell, there are four main gates which are mentioned as follows:

- **Forget Gate:** Responsible for deciding what to retain and what to discard from the previous cell state (C_{t-1}). It prevents irrelevant information from affecting the current state, addressing issues like the vanishing gradient problem. The gate's output, f_t, ranges between 0 and 1, determining what should be remembered and forgotten.

  Equation for the forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

- **Input Gate:** Regulates the introduction of new information into the memory state (C_t). It consists of the input modulation gate, i_t, which decides what values should be updated, and the candidate update, $C_t^{\sim}$, which represents new proposed values. The gate plays a crucial role in determining what information is worthy of being stored and what should be ignored.

Equation for the input modulation gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

Equation for the candidate update:

$$C_t^{\sim} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{11}$$

- **Cell State:** Serves as a central repository for information, continuously updated by the forget and input gates. It blends past knowledge with new candidate values ($C_t^{\sim}$) to make informed decisions about what to keep and what to discard.

  Equation for updating the cell state:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t^{\sim} \tag{12}$$

- **Output Gate:** Determines the relevance of information from the cell state (C_t) for generating predictions or the final output. It acts as a filter, controlling which parts of the cell state contribute to the hidden state (h_t) and, consequently, the output or prediction. The gate's output, o_t, ranges between 0 and 1, allowing selective information flow.

  Equation for the output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{13}$$

  Additional equation to compute the hidden state:

$$h_t = o_t \cdot \tanh(C_t) \tag{14}$$

Bidirectional Long Short-Term Memory

The Bidirectional Long Short-Term Memory (Bi-LSTM) is a variant of the standard LSTM model that captures dependencies in both forward and backward directions. A forward-end LSTM processes the input sequence from the beginning to the end of the layer, while the backward-end LSTM process the sequence from the beginning to the end. The outputs are then concatenated. Bi-LSTM has the same gating mechanisms as standard LSTM models, including the forget gate, input gate, cell state gate, and output gate. By analyzing sequences bidirectionally, it captures contextual information effectively, so it can be used for tasks like natural language understanding and time series forecasting, where both past and future information are essential [45].

Gated Recurrent Unit

As a RNN architecture, the gated recurrent unit (GRU) addresses the limitations of traditional RNNs. In the same way as LSTM, GRU is geared toward sequential data processing, but its structure is more simplified, making it computationally efficient. The core idea behind the GRU is to maintain and update a hidden state that captures essential information from past time steps. In contrast to the LSTM, the GRU has two gates: the update gate and the reset gate. The GRU can adjust its memory efficiently and control the flow of information effectively with these gates [46].

- **Update Gate:** This gate determines how much of the previous hidden state should be retained and how much of the current candidate state should be added. By balancing remembering and forgetting, it allows the model to selectively update its memory.
- **Reset Gate:** This gate determines which information from the previous time step should be forgotten or reset. It aids the GRU model in learning which past information is irrelevant to the current time step.

Bidirectional Gated Recurrent Unit

The Bidirectional Gated Recurrent Unit (Bi-GRU) follows a similar principle to Bi-LSTM, but uses GRU as its core units. GRUs have two main gates, the reset gate and the update gate, which facilitate the flow of information. A Bi-GRU is composed of two sets of GRU layers - one processing the sequence from beginning to end (forward GRU) and one from beginning to end (backward GRU). A comprehensive representation of the input sequence is obtained by concatenating the outputs from both directions. Similar to Bi-LSTM, Bi-GRU captures bidirectional dependencies effectively, but with a simpler architecture due to a reduced number of gates. The Bi-LSTM and Bi-GRU models are both used widely for sequence-to-sequence tasks, including time series forecasting, due to their ability to capture complex patterns and dependencies [47].

4.2.4 Data Preprocessing

The dataset produced for this study consists of 40,320 observations, which corresponds to data collected at one-minute intervals over a 28-day period. The data was collected at one-minute intervals to ensure a high level of detail in capturing the behavior of the battery system. To assess the impact of different time intervals on forecasting accuracy, the dataset was resampled into three subsets: 15-minute, 30-minute, and 1-hour intervals. This resampling was done to make a balance between data granularity and computational efficiency, as the original dataset contained a large volume of one-minute interval observations. Based on the resampled datasets, time series forecasting models were evaluated and results were compared across different temporal resolutions. The subsets consisted of 2,688 observations (15-minute interval), 1,344 observations (30-minute interval), and 672 observations (1-hour interval). This approach enables a comprehensive analysis of the dataset and provides insight into the effect of temporal granularity on forecasting accuracy. An exploratory data analysis (EDA) is performed to preprocess the dataset and select the important features to include in the SoC forecasting model. In Figure 4.6, the correlation heatmap, which is based on Pearson correlation coefficients, clearly shows that the battery voltage and SoC are positively correlated, with a correlation coefficient of 1. This high correlation is not surprising, considering that the MPPT controller relies on the battery voltage as a crucial factor in determining the SoC.

According to the correlation values, the SoC is more correlated with load voltage, solar voltage, and load current. Therefore, these features become crucial factors to consider in order to improve the SoC forecasting model. To further improve the feature selection process, mutual information (MI) is employed alongside correlation analysis. While correlation focuses on linear relationships between

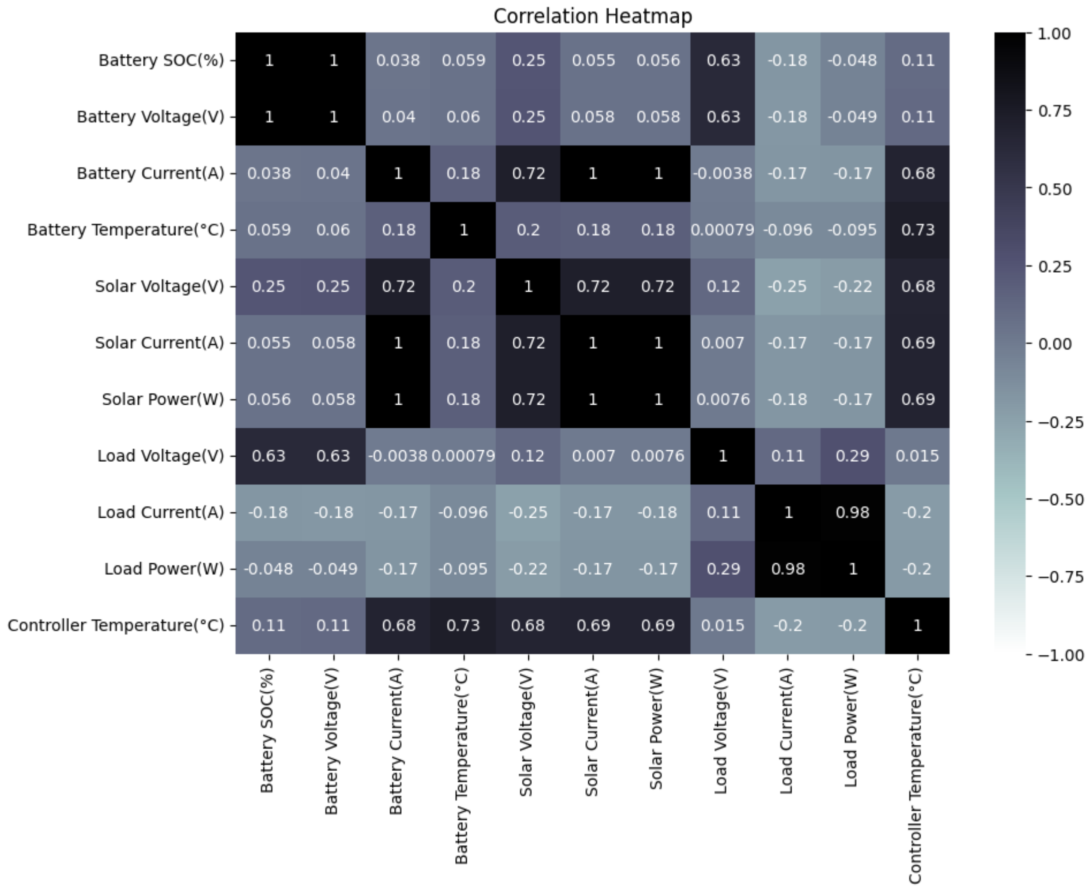

Figure 4.6: Pearson correlation heatmap displaying the relationship between the gathered features and the SoC of the battery

variables, MI takes into account both linear and nonlinear dependencies. It quantifies the amount of information one variable provides about another, capturing a broader range of relationships beyond what correlation alone can reveal. The MI is calculated as:

$$\mathrm{MI}(X, Y) = \sum \sum P(X, Y) \log_2 \left(\frac{P(X, Y)}{P(X) \cdot P(Y)} \right) \tag{15}$$

where $P(X, Y)$ represents the joint probability distribution of variables X and Y, and $P(X)$ and $P(Y)$ represent their marginal probability distributions [48]. Normalized mutual information (NMI) is used to standardize evaluations across feature scales. A high NMI score indicates strong dependence between the target feature and the input feature. NMI scores range from 0 to 1, with 1 representing perfect correlation (0 = no mutual information). Figure 4.7 shows NMI scores between all the features and battery SoC. Battery voltage, solar voltage, load voltage, and battery current have the highest NMI scores, all with NMI score above 0.2. Therefore, based on both analyses, the following features can be considered as features for improving the SoC forecasting model: battery voltage, load voltage, solar voltage, load current, and battery current.

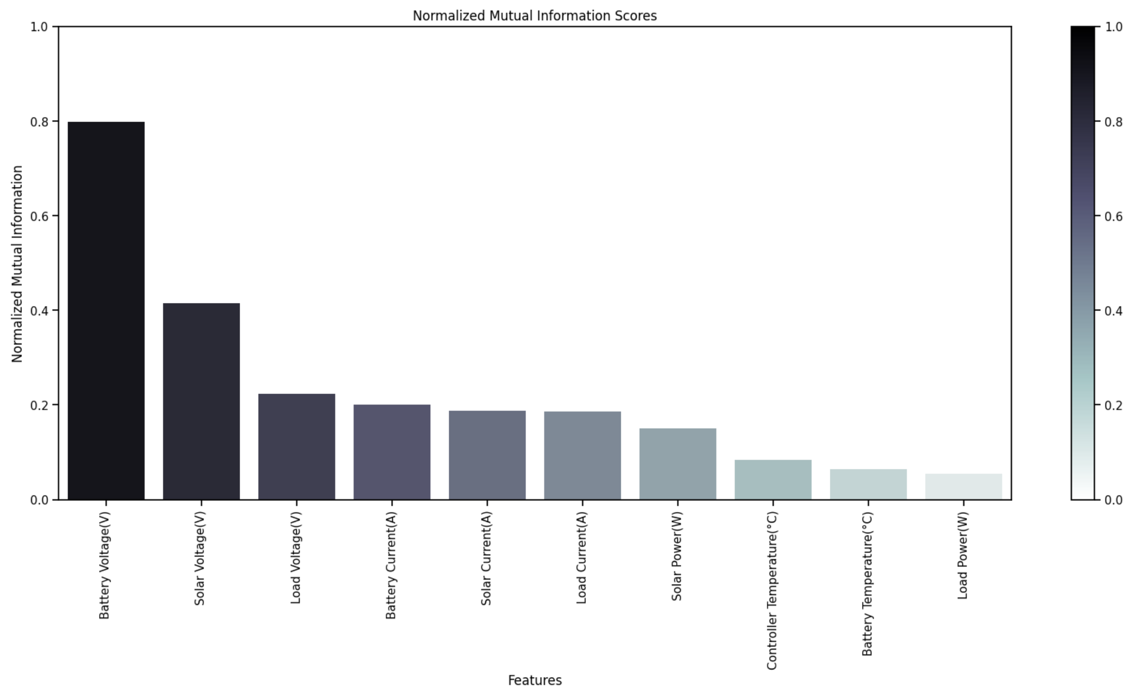

Figure 4.7: Normalized mutual information score between SoC and the features

To enhance our SoC forecasting models, we expanded our feature selection beyond the data collected by the MPPT controller. In addition to the selected variables, we integrated three additional features: solar radiation data, subsystem activation counts, and a categorical time-of-day variable. Solar radiation data accommodates energy input variations, while subsystem activation counts reveal the device's operational status, enriching our forecasting context. The time-of-day feature categorizes data into morning, afternoon, evening, and night based on available daylight for device charging and discharging

In the experiment, the Date and Time were set as the time series index. The SoC percentage and the features selected based on the Pearson correlation and NMI score and additional features were used as input data. The dataset was then divided into training and testing sets, with proportions of 70% and 10%, respectively. Additionally, 20% of the data was allocated for hyper-parameter validation during model development. Furthermore, MinMax scaling with a 0 to 1 range was used to normalize the dataset. To handle the categorical feature of time-of-day, we employed one-hot encoding. This method involved creating binary dummy variables for each category within the time-of-day feature, allowing us to represent it in a way suitable for our modeling process.

4.2.5 Model Development

Various multi-step and multivariate time series forecasting models were applied to the resampled datasets with different time intervals to forecast the SoC of the battery. Both machine learning and deep learning modeling approaches were used in this study. Initial benchmarks were established using machine learning models such as decision trees (DT) and random forests (RF). The data was then modeled using deep learning models including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Bidirectional LSTM (Bi-LSTM), and Bidirectional GRU (Bi-GRU), in order to capture complex temporal patterns. Forecast horizons of 2 hours, 5 hours, and 10 hours were used to evaluate the effectiveness of these models. Training and testing of the models were conducted on resampled subsets of data for each forecast horizon. Using this approach, we were able to evaluate model performance over various time intervals and forecast horizons. The look-back window determines the amount of historical data used for forecasting future time steps. In this study, the look-back windows are set to the same duration as the forecast horizons in each model. For instance, in the case of the 15-minute scale dataset, a forecast horizon of 2 hours corresponds to a look-back window size of 8 (4 data points per hour x 2 hours), indicating that the model considers the previous 8 time steps to make predictions. In order to optimize the performance of the forecasting models, hyperparameter tuning was carried out using a grid search approach to identify the most suitable parameter values for the models. The hyperparameters that were tuned include the number of layers, the number of units for each layer, dropout rate, L2 regularization parameter (lambda), activation function, and learning rate. Adam optimizer is used for tuning the developed deep learning models, and the dropout regularization to avoid overfitting. The forecasting models were implemented using Keras 2.6.0 API with Tensorflow 2.12.0 as the backend, within the Python 3.11.4 environment. Figure 4.8 illustrates an example of model optimization results based on the parameters specified in Table 4.3 for the Bi-GRU model. The plot demonstrates the decreasing training and validation losses over multiple epochs, indicating a well-fitted curve. It is evident that the optimal number of epochs for this model is approximately 40, as both losses reach a stable point.

Table 4.3: Parameters set for the models

Parameter	Value
Training portion	70% of total data
Validation portion	20% of total data
Testing portion	10% of total data
Normalization	MinMax normalization (range 0 to 1)
Regularization	Dropout, L2 regularization
Early stop to avoid overfitting	patience = 30
look-back window size range	2-40 data points
forecast horizons	2 hours, 5 hours, 10 hours
Optimization algorithm	Adam
Maximum number of epochs	150
Testing evaluation metrics	MAE and RMSE
Hyperparameter Tuning Method	Grid Search
Tuned Hyperparameters	Number of layers, Number of units per layer, Dropout rate, L2 regularization parameter (lambda), Activation function, Learning rate

4.2.6 Evaluation

Figure 4.9 provides a detailed comparison of the forecasting models used in this study, based on the MAE and RMSE performance metrics. Three resampled datasets with 15-minute, 30-minute, and 1-hour scales are used to evaluate the forecasting models, covering forecast horizons of 2 hours, 5 hours, and 10 hours. The figure displays the errors for each model's last time step forecast, providing insights into the effectiveness of the models for long-term forecasting. It is observed that RMSE errors are generally higher than MAE errors, indicating RMSE's sensitivity to large errors. Additionally, and as expected, errors for all models tend to increase with increasing forecast horizons, reflecting the complexity and uncertainty of long-term forecasts. The Bi-GRU model outperforms other models across various forecast horizons and dataset scales, making it a good choice for more accurate predictions.

Further quantative evaluation of forecasting models was conducted in addition to visual analysis

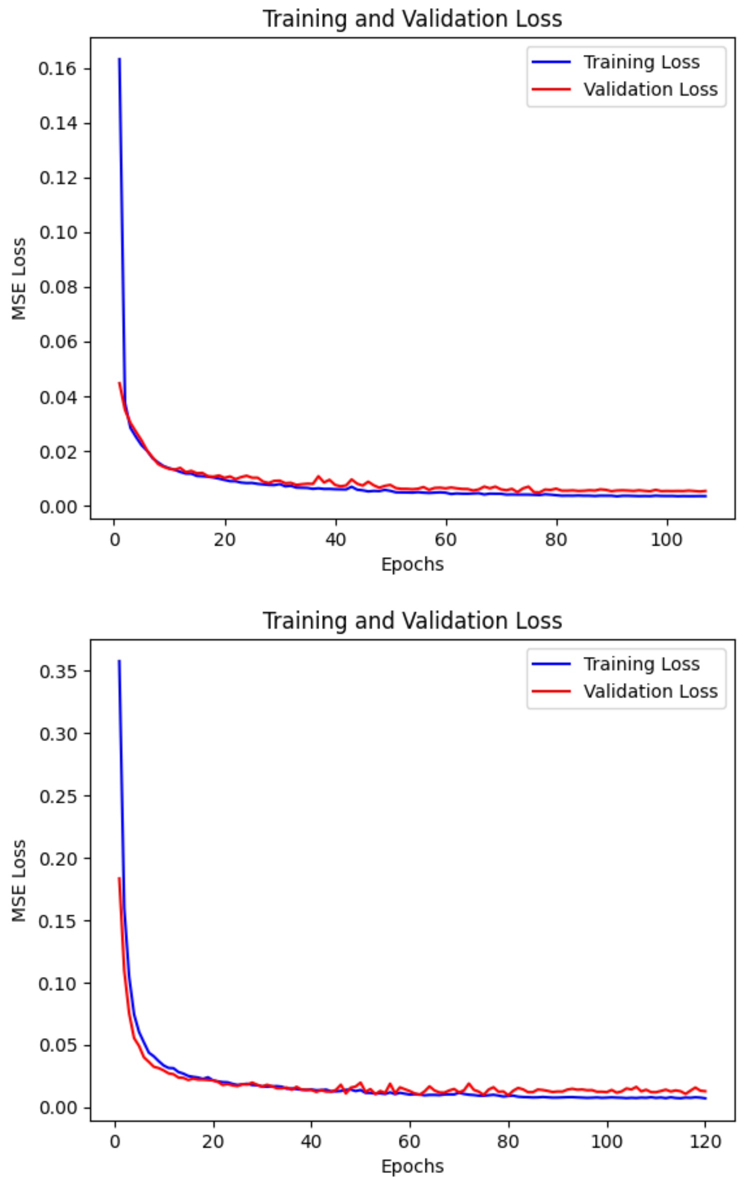

Figure 4.8: Training and Validation Loss Curves for Bi-GRU Model

Figure 4.9: Comparison of forecasting models for the last time step on each dataset

from the figure. While the figure enabled comparison of forecasts for the last time step, Table 4.4 provided an assessment of the models' overall performance. This evaluation involved comparing the average errors for all time steps across the longest forecast horizon of 10 hours. In this way, the assessment allowed for evaluating the model's accuracy and reliability across the entire forecast period, gaining a better understanding of their overall performance and forecasting capabilities. Bi-GRU model has the lowest MAE and RMSE, indicating the highest accuracy among all models. It shows that a bidirectional architecture and gated recurrent units make the Bi-GRU model a proper model for Battery SoC forecasting.

Table 4.4: Overall Forecast performance on test data for the horizon of 10 hours

Models	15 min dataset		30 min dataset		1 hour dataset	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
DT	6.02	7.33	4.76	6.91	6.20	7.46
RF	4.49	5.71	3.56	5.36	4.70	5.84
CNN	4.45	5.60	3.91	4.92	4.39	5.59
LSTM	4.63	5.34	3.88	4.69	4.24	5.46
GRU	4.46	5.30	3.71	4.55	3.39	4.34
Bi-LSTM	3.88	4.69	3.70	3.42	2.93	3.72
Bi-GRU	**3.61**	**4.57**	**2.55**	**3.26**	**2.79**	**3.66**

4.3 Power Management Service Levels

In developing an effective power management system for the SCiNe device, defining distinct service levels that optimize power consumption and functionality is a key consideration. Service levels are designed to achieve a balance between maintaining the device's core functions and extending its operational life. The power management system operates as a decision-making framework, allowing the SCiNe device to adapt to varying energy conditions and prioritize its tasks accordingly. To achieve this, the system leverages battery State of Charge (SoC) forecasting, adjusting service levels in real-time to ensure an optimal balance between functionality and power conservation.

Three service levels can be implemented in the power management system, each tailored to specific SoC ranges. These service levels are:

- **Normal Mode :** Normal Mode allows the device to operate at full functionality, providing all its services and features. It is activated when the SoC is within a range of 100% to 60%. In Normal Mode, the device uses all of its energy resources.

- **Power Saving Mode :** SCiNe enters Power Saving Mode when the SoC falls between 59% and 20%. This mode optimizes energy consumption by activating various power-saving mechanisms. LEDs are dimmed by 50%, the device reduces data transmission frequency to send telemetry data to the cloud, screen updates ocurr less frequently, conserving energy while maintaining essential functions.

- **Emergency Mode :** The Emergency Mode is activated when the SoC reaches 19% to 1%. In Emergency Mode, the device prioritizes essential functions, like displaying the bus schedule, while conserving energy by dimming LEDs to 25% brightness and disabling subsystems like the speaker, microphone, and people counting module. In Emergency Mode, the device focuses on preserving energy to extend its operational duration and provide essential services.

As a result of implementing these service levels, the SCiNe device can dynamically adapt to changing energy conditions and operational requirements. As a result, the SCiNe device operates optimally while maximizing energy efficiency, enhancing its reliability and longevity. As part of the seamless transition between these service levels, battery SoC forecasting plays a crucial role, ensuring that the device can respond effectively to fluctuations in its energy supply regardless of the level of service.

Chapter 5

Conclusions and Future Work

The integration of Artificial Intelligence (AI) and the Internet of Things (IoT) in AIoT-driven solutions presents innovation, but also energy challenges. To tackle these challenges while adhering to the principles of responsible AI, it is crucial to prioritize the sustainability of these solutions through the promising adoption of renewable energy sources. Despite the benefits of renewable energy, challenges such as its intermittent nature necessitate the implementation of an effective power management system. This study focused on developing an effective power management system, serving as a decision-making framework for AIoT solutions. This was done through the development of a power management system for a battery-powered AIoT device charged through a solar panel named "SCiNe", with a specific emphasis on accurate battery State of Charge (SoC) forecasting. To understand the behavior of the system, an experiment was designed and a custom data logging system was developed to gather relevant data, enabling accurate analysis and model development. The study explored the multivariate and multi-step time series forecasting domain, using a variety of models, including decision trees (DT) and random forests (RF), to deep learning models of CNN, LSTM, GRU, Bi-LSTM, and Bi-GRU. The models were evaluated using both last time step forecasts for a comparative view and average errors over the entire forecast period for a comprehensive evaluation. The Bi-GRU model outperformed other models across datasets with varying time intervals and forecast horizons. These findings highlight the potential of the Bi-GRU model for real-world applications in similar systems. Incorporating additional input features, exploring alternative time series forecasting models, and integrating the SoC forecasting solution into the decision-making system, offer promising avenues for future enhancements in the study. This study has primarily addressed the first phase of the decision-making system for managing AIoT device power – accurate battery SoC forecasting. The next step is to design and implement control strategies that enable dynamic adjustments to service levels of the device. These service levels define specific operating modes for the device, with each level corresponding to different functionalities and power consumption limits, ensuring both system stability and power efficiency. These enhancements lead to the development of a sustainable power management system for AIoT applications.

Appendix A

Setup Configuration

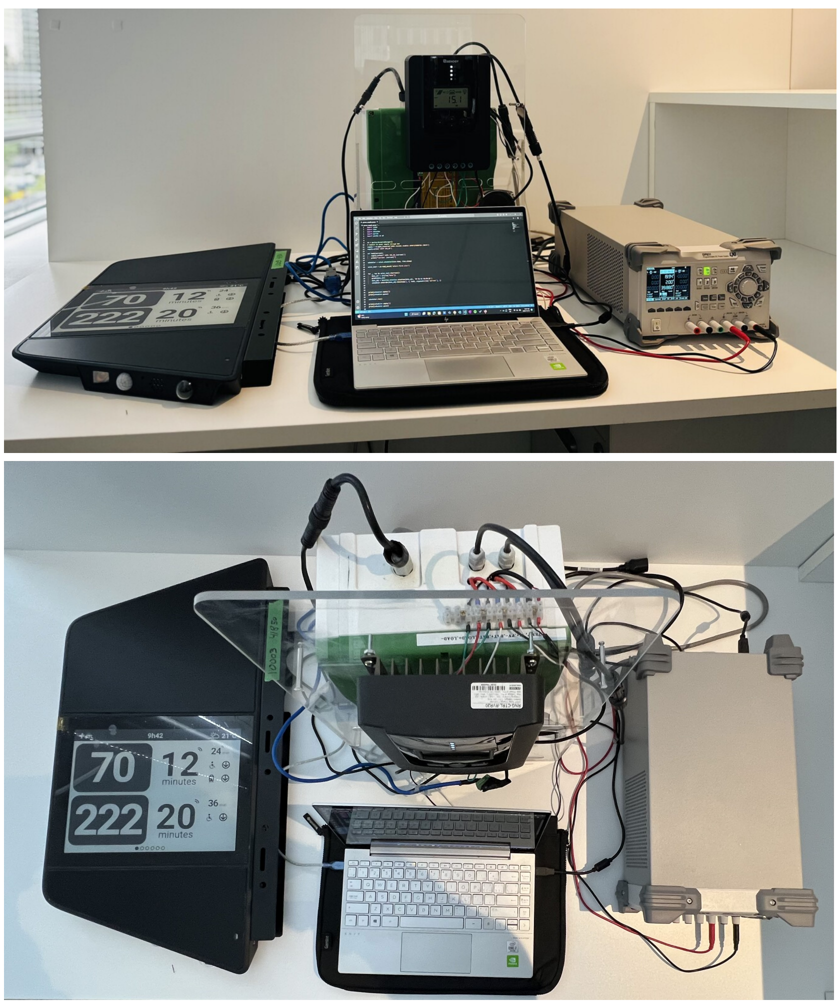

Figure A.1: Setup Configuration

Bibliography

[1] Feng Xia, Laurence T Yang, Lizhe Wang, Alexey Vinel, et al. Internet of things. *International journal of communication systems*, 25(9):1101, 2012.

[2] Heather Richter Lipford, Madiha Tabassum, Paritosh Bahirat, Yaxing Yao, and Bart P. Knijnenburg. *Privacy and the Internet of Things*, pages 233–264. Springer International Publishing, Cham, 2022.

[3] Lorik Fetahu, Arianit Maraj, and Abdullah Havolli. Internet of things (iot) benefits, future perspective, and implementation challenges. *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 399–404, 2022.

[4] Rajalakshmi Krishnamurthi, Adarsh Kumar, Dhanalekshmi Gopinathan, Anand Nayyar, and Basit Qureshi. An overview of iot sensor data processing, fusion, and analysis techniques. *Sensors*, 20(21), 2020.

[5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[6] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2021.

[7] Youcef Guenfaf and Youcef Zafoune. An iot ml-based system for energy efficiency in smart homes. In *2023 IEEE World AI IoT Congress (AIIoT)*, pages 0198–0203, 2023.

[8] Thaier Alawadh and Salem Alzahmi. Solar energy adoption: Technical advantages and challenges for the gulf cooperation council countries. *2021 6th International Conference on Renewable Energy: Generation and Applications (ICREGA)*, pages 184–188, 2021.

[9] Muralidhar Nayak Bhukya, Manish Kumar, Akshat Kant, and Punit. Renewable energy: Potential, status, targets and challenges in rajasthan. *Journal of Physics: Conference Series*, 1854, 2021.

[10] Kamran Dawood. An overview of renewable energy and challenges of integrating renewable energy in a smart grid system in turkey. *2020 International Conference on Electrical Engineering (ICEE)*, pages 1–6, 2020.

[11] Asmita Ajay Rathod and Balaji Subramanian. Scrutiny of hybrid renewable energy system for control, power management, optimization and sizing: Challenges and future possibilities. *Sustainability*, 2022.

[12] Seyed Mohsen Hosseini, Raffaele Carli, and Mariagrazia Dotoli. Robust optimal energy management of a residential microgrid under uncertainties on demand and renewable power generation. *IEEE Transactions on Automation Science and Engineering*, 18:618–637, 2021.

[13] Anindya Bharatee, Pravat Kumar Ray, Bidyadhar Subudhi, and Arnab Ghosh. Power management strategies in a hybrid energy storage system integrated ac/dc microgrid: A review. *Energies*, 2022.

[14] Abdellatif Elmouatamid, Radouane Ouladsine, M. Bakhouya, Najib EL KAMOUN, Khalid Zine-dine, and Mohammed Khaidar. Mapcast: an adaptive control approach using predictive analytics for energy balance in micro-grid systems. *International Journal of Renewable Energy Research*, 10:945, 07 2020.

[15] Youssef NaitMalek, Mehdi Najib, Mohamed Bakhouya, and Mohamed Essaaidi. Embedded real-time battery state-of-charge forecasting in micro-grid systems. *Ecological Complexity*, 45:100903, 2021.

[16] Seongyun Park, Jeongho Ahn, Taewoo Kang, Sungbeak Park, Youngmi Kim, Inho Cho, and Jonghoon Kim. Review of state-of-the-art battery state estimation technologies for battery management systems of stationary energy storage systems. *Journal of Power Electronics*, 20(6):1526–1540, Nov 2020.

[17] Hongwen He, Rui Xiong, Hongqiang Guo, and Shuchun Li. Comparison study on the battery models used for the energy management of batteries in electric vehicles. *Energy Conversion and Management*, 64:113–121, 2012. IREC 2011, The International Renewable Energy Congress.

[18] M.A. Hannan, M.S.H. Lipu, A. Hussain, and A. Mohamed. A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations. *Renewable and Sustainable Energy Reviews*, 78:834–854, 2017.

[19] Yinsheng Luo Qi Wang and Xiaoxin Han. Research on estimation model of the battery state of charge in a hybrid electric vehicle based on the classification and regression tree. *Mathematical and Computer Modelling of Dynamical Systems*, 25(4):376–396, 2019.

[20] M. S. Hossain Lipu, M. A. Hannan, Aini Hussain, Shaheer Ansari, S. A. Rahman, Mohamad H.M. Saad, and K. M. Muttaqi. Real-time state of charge estimation of lithium-ion batteries using optimized random forest regression algorithm. *IEEE Transactions on Intelligent Vehicles*, 8(1):639–648, 2023.

[21] Bharath Pattipati, Chaitanya Sankavaram, and Krishna Pattipati. System identification and estimation framework for pivotal automotive battery management system characteristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):869–884, 2011.

[22] Lin Hu, Xiaosong Hu, Yunhong Che, Fei Feng, Xianke Lin, and Zhiyong Zhang. Reliable state of charge estimation of battery packs using fuzzy adaptive federated filtering. *Applied Energy*, 262:114569, 2020.

[23] Juan Carlos Álvarez Antón, Paulino José García Nieto, Cecilio Blanco Viejo, and José Antonio Vilán Vilán. Support vector machines used to estimate the battery state of charge. *IEEE Transactions on Power Electronics*, 28(12):5919–5926, 2013.

[24] ShuMei Zhang, Lin Yang, XiaoWei Zhao, and JiaXi Qiang. A ga optimization for lithium–ion battery equalization based on soc estimation by nn and flc. *International Journal of Electrical Power Energy Systems*, 73:318–328, 2015.

[25] Prashant Shrivastava, Tey Kok Soon, Mohd Yamani Idna Bin Idris, and Saad Mekhilef. Overview of model-based online state-of-charge estimation using kalman filter family for lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, 113:109233, 2019.

[26] Ardiansyah, Yeonghyeon Kim, and Deokjai Choi. Lstm-based multi-step soc forecasting of battery energy storage in grid ancillary services. In *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 276–281, 2021.

[27] Ardiansyah, Zaki Masood, Deokjai Choi, and Yonghoon Choi. Seq2seq regression learning-based multivariate and multistep soc forecasting of bess in frequency regulation service. *Sustainable Energy, Grids and Networks*, 32:100939, 2022.

[28] Youssef NaitMalek, Mehdi Najib, Anas Lahlou, Mohamed Bakhouya, Jaafar Gaber, and Mohamed Essaaidi. A hybrid approach for state-of-charge forecasting in battery-powered electric vehicles. *Sustainability*, 14(16):9993, Aug 2022.

[29] Y. NaitMalek, M. Najib, M. Bakhouya, and M. Essaaidi. On the use of machine learning for state-of- charge forecasting in electric vehicles. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 408–413, 2019.

[30] Aleksei Mashlakov, Samuli Honkapuro, Ville Tikka, Arto Kaarna, and Lasse Lensu. Multi-timescale forecasting of battery energy storage state-of-charge under frequency containment reserve for normal operation. In *2019 16th International Conference on the European Energy Market (EEM)*, pages 1–8, 2019.

[31] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[32] Lambros Mitropoulos, Annie Kortsari, Vasilis Mizaras, and Georgia Ayfantopoulou. Mobility as a service (maas) planning and implementation: Challenges and lessons learned. *Future Transportation*, 3(2):498–518, 2023.

[33] Jiju Antony and Cahyono St. *Design of Experiments for Engineers and Scientists SECOND EDITION*. 07 2022.

[34] Visual Crossing Corporation. Visual crossing weather api, 2023. Accessed on March 01, 2023.

[35] Yinjiao Xing, Wei He, Michael Pecht, and Kwok Leung Tsui. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Applied Energy*, 113:106–115, 2014.

[36] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):160, Mar 2021.

[37] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[38] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006.

[39] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[40] Uppala Meena Sirisha, Manjula C. Belavagi, and Girija V. Attigeri. Profit prediction using arima, sarima and lstm models in time series forecasting: A comparison. *IEEE Access*, 10:124715–124727, 2022.

[41] Marco Peixeiro. Time series forecasting in python, 2022.

[42] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, volume 138. 01 2013.

[43] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[44] Christopher Olah. Understanding lstm networks, 2015. Accessed on 2023-11-06.

[45] Hanlei Sun, Jianrui Sun, Kun Zhao, Licheng Wang, and Kai Wang. Data-driven ica-bi-lstm-combined lithium battery soh estimation. *Mathematical Problems in Engineering*, 2022:9645892, Mar 2022.

[46] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. 12 2014.

[47] Xuechen Li, Xinfang Ma, Fengchao Xiao, Cong Xiao, Fei Wang, and Shicheng Zhang. Time-series production forecasting method based on the integration of bidirectional gated recurrent unit (bi-gru) network and sparrow search algorithm (ssa). *Journal of Petroleum Science and Engineering*, 208:109309, 2022.

[48] Hongfang Zhou, Xiqian Wang, and Rourou Zhu. Feature selection based on mutual information with correlation coefficient. *Applied Intelligence*, 52(5):5457–5474, Mar 2022.