Applying a Machine Learning Model to Estimate the Current State of Charge of Energy Storage Devices

Yasmin Salce (s1530828)
February, 2022

Supervisors:
Dr. Ir. G Hoogsteen
Dr. Ir. B. Homan
DR. Ir. A.B.J. Kokkeler
Dr. Ir. M.E.T Gerards
Faculty of Electrical Engineering,
Mathematics and Computer Science
Department of Computer Architecture
and Embedded Systems
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

This research examines the role of applying machine learning to estimate the state of charge (SoC) of energy storage devices. SoC is a vital parameter that can be used to reflect the performance of the energy storage device and is key in the management of these energy storage systems, such as batteries or a hot water buffer. It is used to optimize the performance and to extend the lifetime of storage systems. A simple artificial neural network (ANN) and a time-series long short term memory (LSTM) neural network is investigated to create a model that can be used on multiple forms of energy storage devices. The simple ANN achieved results of over 80% accuracy for the Yuasa and Multipower batteries but did poorly at 25% accuracy for the Conrad battery. The LSTM achieved an accuracy of over 90% for 4 batteries and a hot water buffer. The LSTM model is then altered so that it is able to estimate the SoC in real time for the Conrad battery, this achieved a lower accuracy of 70.1%.

List of Abbreviations

AI Artificial Intelligence
ANN Artifical Neural Network
BP Backwards Propogation
DiBu Diffusion Buffer Model
FF Feedforward
GRU Gated Recurrent Unit
LiFePo Lithium Iron Phosphate battery
Li-ion Lithium-ion Battery
LSTM Long Short Term Memory
MAE Mean Absolute Error
ML Machine Learning
MPC Model Predictive Control
MSE Mean Squared Error
Ni-Cd Nickel-cadmium Battery
Ni-MH Nickel-metal Hydride Battery
NN Neural Network
RBF Radial Bias Function
RL Reinforcement Learning
RMSE Root Mean Squared Error
RNN Recurrent Neural Network
SoC State of Charge
SVM Support Vector Machine

Table of contents

1 Introduction

In the last few years, the effects of climate change are becoming more noticeable. With this awareness, there has been an increase into the development of sustainable energy and energy storage systems. The trend is to move away from fossil fuels and find alternative solutions for energy. The issue with fossil fuels are the emissions, that contribute to climate change. Policies and regulations play a big role in this energy transition from fossil fuels to sustainable energy, most noticeably the Paris Agreement [1]. To meet the goals presented in the Paris Agreement, the energy landscape needs to change and limit the use of fossil fuels.

Currently in the Netherlands, renewable energy makes up 5.8% of all energy sources, while natural gasses and oil making up 79.2% [2]. Smart grids can be used in collaboration with energy storage systems. A smart grid system is an electrical grid that is able to monitor and control the energy 'from the points of generation to customers in a smart way' [3]. However, issues presents itself with storing energy, such as the amount of energy the grid needs, the consumption or the rate of decay, and many other factors. The issue with grids is that the supply and demand of energy may not be balanced, which can cause problems such as power outages [3]. These issues can be critical, therefore the use of energy management systems becomes highly important. There are parameters that need to be monitored to control the smart grid. The state of charge (SoC) being one of them. When the SoC is known, charge cycles can be managed better. This results in improved management of systems and extended battery life.

SoC estimation is a fundamental challenge. It is difficult to estimate the state of charge for various kinds of systems in different domains, such as in the electrical or thermal domain. Example of storage devices in these domains would be a battery or hot water buffer. This difficulty is due to the non-linear characteristics of these devices, particularly in batteries, that also includes complex electrochemical reactions. The SoC is a critical parameter that signifies the available charge of an energy storage device.

Through the use of machine learning (ML), the grid is able to make smart decisions and respond to changes in the system, such as the changes in customer demands, changes in energy output and power outages. Due to the difficulty of these factors, Machine learning algorithms are being applied and can provide solutions for dealing with the problems presented.

ML is becoming a more prominent component in the energy industry. ML is a subset of artificial intelligence (AI). AI solves tasks that require human intelligence, and ML solves tasks by learning from data and making estimations and predictions. ML allows computers to learn and is the process of learning from the data provided, by discovering patterns and gathering insight about the data. AI enabled energy storage can aid in collecting and analysing the data. AI has the capability to solve various issues, regarding an energy management system, accurately estimating the state of charge [4].

ML is an efficient method for analyzing and processing large amounts of data. Businesses that are able to benefit from ML, by analyzing their data and gaining insights. Since ML is a promising technique for the energy sector, it is rapidly being used by several industries and is predicted to grow to '8.81 billion USD by 2022, at a compound annual growth rate of 44.1%' [5], showing that ML will be used more though out the industries. ML algorithms will be able to drastically advance the energy storage sector.

New types of ML will continue to emerge and create possibilities for exploration [6]. Simulations based on ML based models can be used to predict e.g. failures, the state of charge (SoC) and to optimize power utilisation.

1.1 Problem Statement

The main issue when it comes to smart grids is its communication and control strategy, which must be addressed in order to achieve the benefits. When there is an issue with communication, the smart grid will not be able to function effectively. SoC is one of the factors involved in the control strategy, which will allow people to better control their energy use. The issue with SoC and batteries in particular, is that many different models exist that are only applicable to certain kinds of batteries. Another issue with these models is that they require complex calculations that are too complicated for real world applications, since they require a large number of parameters. These models work best in a test environment where the variables and processes can be closely monitored. Kazmi and Suykens [7] explain that methods employed in the design phase, rarely reflect operational performance of the modelled systems, due to unexpected behaviour.

On the other hand, models can also be too simple and deviate too much from the real world to be useful. Thus, these models can be deemed useless because of the differences and complexity. However, the DiBu model [8] is designed for multiple energy storage devices, and simple so that is is suitable for real world applications. Due these two problems, there is an increasing interest in applying various machine learning techniques to estimate the SoC and compare it to existing models. Current methods tackle many kinds of machine learning algorithms, such as; Kalman filters, particles filter, support vector machines and neural networks. Using ML models could potentially reduce the number of parameters required for calculating the SoC and thus simplify the estimation process.

The goal is to research machine learning algorithms and investigate whether they can yield a good result in the state of charge estimation of storage devices. These results will aid in smart grid control and efficiency and help with energy management systems. Current models and algorithms exist for this for specific energy storage devices, but to the best knowledge of the author, none have developed an abstract model that can be used on various kinds of storage devices.

1.2 Research Questions

There are two main objectives within this project, which will lead to the research questions. They are:

- **Research Objective 1** Apply a machine learning algorithm that is able to accurately estimate the state of charge of a storage device.
- **Research Objective 2** Ensure that the algorithm can be used to estimate the state of charge of various kinds of storage devices, with minimal tuning.

Based on the objectives of the project, the main research question is formulated as follows:

- What is the trade-off of applying machine learning for state of charge estimation of energy storage devices compared to applying existing complex models?

In order to answer the main research question, four sub-questions have been defined and are presented below:

- What existing model/domain knowledge is already present and how do they compare?
- What measurable variables are important for the generalisation of the model, that are applicable to storage in general?
- Which machine learning technique is the most applicable for estimating state of charge?
- Which evaluation metrics and methods allow for a comparative study?

1.3 Section Overview

The report will continue with the following structure. Chapter 1 provides a brief introduction of the problem at hand and motivates the necessity of exploring the use of machine learning as a tool to estimate the state of charge. Chapter 2 presents the related work, it provides existing state-of-art techniques used to solve similar problems and introduces current SoC estimation approaches such as impedance, model based and data-driven methods. The evolution of the estimation methods are presented, where machine learning is becoming more prominent for this task.

Chapter 3 provides the research methodology, which is the theoretical way in which the models can be implemented. It is the proposed approach of the research, as well as a description about the data set. In the proposed methodology, algorithms are implemented for basic estimation to provide information on how to proceed. The evaluation metrics are described to provide insight into how accurate the final model is.

Chapter 4 displays the results based on the method and implementation of Chapter 3. Chapters 5 and 6 are the discussion and conclusion.

2 Related Work

This chapter presents relevant literature with respect to the aims and goals of this research.

2.1 Storage Systems

There are many types of energy storage systems, such as, thermal, chemical and electrical storage. Energy storage systems are systems that store energy at a certain time, so it can be used at a later time. This principle can be used to make an energy system more balanced. These energy storage systems are used in smart grids and their planning. A smart grid can be defined as: 'a combination of a traditional distribution network and a two-way communication network for sensing, monitoring, and dispersion of information on energy consumption' [9]. It is able to integrate multiple energy sources into the grid and is able to share and exchange energy between the energy sources available [3].

Storage devices are found in different energy domains, that can be modeled in a similar way. For example, there are similarities between a hot water buffer and a battery, as they are analogous. These storage types work in the thermal and electrical domain respectively, but have comparable parameters. The pressure and the flow of the hot water buffer, can be compared to the voltage and the current of the battery. Gong et. al. [10] proposed a generalised energy storage model in their research for a battery and a water heater. From this research it can be concluded that, energy storage can be generalised for the different domains. The generalisation of the system can be modeled and all internal reactions can be neglected, in order to create a model that will be applicable across multiple domains [10].

The SoC for energy storage, is used to determine the remaining capacity and is a key aspect when it comes to the control process. It will be able to reflect the battery performance. Which will allow it to protect the battery, prevent over-discharging and thus improving the battery life by allowing applications to make rational control strategies for energy saving [11]. In order to make more accurate predictions, ML can be used. ML will be able to improve the accuracy of the SoC predictions and capture complex calculations by training and learning to improve its accuracy [12]. Due to this, it is suitable for the real-time management of the system and will be explored in Section 2.3.

2.2 Control strategies

The SoC is an important parameter for control strategies as this information allows for control decisions to be made by the application. In order to save energy for battery management. Control algorithms for smart energy systems need SoC information 'to predict the amount of storage capacity left for charging or discharging in the near future' [13]. For example, Homan and van Leeuwen [13] use a predictive model that predicts the minimum SoC and the electric power consumption for smart control. Accurate estimation models are needed with these control systems due to the resulting effects, such as energy usage and power flow [8].

Another method, such as reinforcement algorithms have been applied for Model Predictive Control (MPC) [14]. Ruelens et. al. and Patyn [15] [14] use a design process for smart control of an electric water heater to find a control policy. Their controller is able to reduce the costs of the total energy consumption by 15% compared to a thermostat. The steps of their process involved; select a model, estimate parameters, estimate the state of the system and, forecast variables [15]. Reinforcement learning (RL) is a field of machine learning, that uses artificial agents to perform actions and studies how it reaches its goals. RL techniques can be simplified to learning a control policy that is able to interact with its environment without modeling [15].

2.3 State of Charge Estimation Methods

There are many developments when it comes to SoC estimation methods where mathematical principles are used. In literature, most SoC models are designed for batteries, and a few for hot water buffers. The most popular batteries being used in the industry are: Ni-Cd battery, Ni-MH battery, lead-acid battery and Li-ion battery. An accurate SoC estimation remains an ongoing issue and can be challenging to implement. SoC methods for battery models can be limited, require too many parameters, have too many uncertainties and may only be used for one specific type of battery [16]. The battery models take into account the internal chemical reactions, such as corrosion, temperature and gassing [16]. Therefore, battery specific models cannot be entirely used to create a generalised model. The diffusion buffer model (DiBu-model) from Homan [8] is able to produce accurate results for multiple types of batteries and can therefore be broadly applied. This is proven by having a less than 5% difference in measured results compared to the predicted results.

There are trade-offs regarding SoC estimation are: simplicity, accuracy, speed and applicability. A high SoC accuracy, such as Husnayin's method, requires elaborate calculations, computations and a large amount of data to yield a good result [16]. To contrast, the Kinetic Battery Model (KiBaM), is able to estimate the SoC with only three variables using non-linear equations [17]. Therefore, the main issue with a high estimation accuracy is the complex method that is required for calculations.

The methods that have been used to estimate SoC are: open circuit voltage, coulomb counting, model based observer, data driven models and impedance based models. Two of these methods are not necessary for this research. The open circuit voltage method, is used just for batteries. Coulomb counting finds the SoC by integrating the current over time and can identify the total sum of energy after requiring an initial SoC [18]. The method produces highly accurate results. However, it must be in a controlled environment where uncertainties can be controlled and therefore only used in a laboratory [18].

Impedance method

The impedance method is a form of direct measurement. The impedance parameters for SoC can be same for different types battery systems, but even with this information, more experiments need to be done in order to identify the best parameters for estimating the SoC of a particular storage system [19]. This method will not be a good estimation of SoC as the battery ages as the error accumulates [18]. Just like Coulomb Counting, this method is better suited to a lab setting rather than in practice and is specific to battery types.

Model based

There are different model based methods that can be used depending on the problem. For example, the paper by Liao [20] explains that the equivalent circuit model approaches that involve lithium are not suitable for SoC estimation due to unknown variables and nonlinear equations. The author of this paper further comments that electrochemical impedance models are highly sensitive and therefore not appropriate for SoC estimation outside a lab setting. Chang [19] on the other hand, argues that the equivalent circuit model can be used for online applications for SoC estimation, which is possible due to the low computational effort, but the trade off results in a limited accuracy. Due to this, there is a prominent trade-off between accuracy and computational efficiency.

Model based techniques that have feedback mechanisms are able to correct possible biases. Another method is the Kalman filter, these filters have been heavily researched and have shown a reasonable accuracy in their estimation. The filters have a robust performance, making them the optimum state estimator for these systems [18]. Section 2.4 further explains Kalman filters. A predictive model for thermal storage [21] uses a charging and discharging model to determine several variables. These variables are the initial SoC, current SoC and the energy that it is able to supply. The model presented used simple algebraic equations that were easy to evaluate. The errors were $1-9\%$ for the charging cycle and $5-9\%$ for the electrical power consumption.

Data driven

Data driven models include machine learning models. Machine learning techniques, such as; neural networks, support vector machines (SVM), linear regression and transfer learning are used to analyze the relationship between the inputs (eg. current, voltage) and its output (eg. SoC). Kazmi and Suykens [7] use a transfer learning approach to model the hot water systems of 61 houses. They demonstrated that this could improve the accuracy of a hybrid method and that depending on the quantity of data and devices, this method can yield great performance gains. With an abundance of data and computational advancements, machine learning techniques are becoming more prevalent and are providing researchers further advancements in their work [22].
Machine learning techniques with regard to their SoC estimation will be evaluated in Section 2.4.

2.4 Machine Learning Techniques

As mentioned in the Introduction, machine learning is a subset of artificial intelligence. The differences between the two is that AI is based on mimicking human behaviour, whilst machine learning is about training a machine so that it has the capability to learn.

There are two types of machine leaning, supervised and unsupervised. Supervised learning learns based on a labeled data set, which is used for classification and regression [12]. Unsupervised learning learns based on an unlabeled data set and is generally used for clustering [12]. Selecting the most appropriate machine learning algorithm can be quite the problem in itself. The algorithm best suited for the task will depend on a few factors such as, the amount of data that is readily available and the quality of the results. Accurate SoC estimation will require a lot of processing power to handle a large data set.

SoC estimation has been used in combination with machine learning many times before, but it has mostly been done for different types of batteries, this section will cover some examples. Machine learning algorithms are evaluated based on their battery estimation complexity, estimation accuracy and what the algorithm is capable of.

Kalman Filter

The Kalman filter method has been able to estimate the SoC for batteries in real time. It does this by using the state of the previous time step and current to make a estimation for the next state. This method is used to filter the input and output and produces an estimate of a system's state [18]. A drawback with the Kalman filter is that the computational load will be high due to the requirement of linearizing around an operating point [23]. The problem with this is that it often requires large amount of parameters, tedious model identification, too many complex equations and can require different versions of the model to perform SoC estimations [23]. All of these factors can increase the computational load. Other than its high computational effort, Kalman filters are vulnerable to aging and temperature [24].

Prashant [20] goes into detail about various types of Kalman filters for lithium-ion batteries and states the common issues amongst the filters. These issues are related to; selecting the various parameters for the the model, tuning the model and selecting the operating conditions. This can all influence the errors that are accumulated, as each filter accumulates errors differently [20]. The author [20] concludes by saying that a Kalman filter can only be used for SoC estimation if combined with another algorithm. Therefore, the biggest fault with the Kalman filter is its computational expense, error accumulation and its dependence on another algorithm to yield the best results.

Particle Filter

Particle filters are similar to Kalman Filters. They are both Bayesian-based estimation algorithms, that have nonlinear and non-Gaussian characteristics [25]. Particle filters use weighted random samples, these sample are seen as the particles, and can be used to estimate the state of the system [26]. These particles are key when it comes to the accuracy and speed of the algorithm, the higher the amount of particles, the higher the speed and accuracy [20]. They are also associated with a large computational demand, which limits its ability to perform for many real-time systems. A particle filter depends on the noise variance for accurate estimations.

Experiments presented in [26] show that the Particle filter and Kalman filter yield similar performance results with regards to estimation accuracy. The particle filter is shown to be six times faster than the Kalman filter, making it more practical for real time systems. Liu et. al. [27] also yielded a high estimation accuracy with a maximum error of 5.1e-2, with a computational time of $4s$.

Support Vector Machine

A SV) uses as a nonlinear estimation system that is insensitive to small changes [28]. The SVM based estimator from Hansen and Wang [28] is able to eliminate the disadvantages of the Coulomb counting method and provide accurate SoC estimations. This method is investigated for a Li-ion battery. SVM is best used when training data is scarce, but this increases computational effort [19]. Hannan et. al. [24] agrees and states that 'SVMs can estimate SoC accurately', but due to the complexity of the computations, it is difficult to execute it in an energy management system.

Neural Networks

Neural networks (NN) are one of the most commonly known ML algorithm used for SoC estimation. The reason being, it is a robust algorithm that can perform under different dynamics, loads, and temperatures. It does not require a mathematical model and can handle non-linear and complex systems. The challenging aspects of a NN is selecting its design features, such as; which activation function to use, how many neurons in hidden layer and the value of the learning rate. A neural network is made up of neurons, each neuron can perform a specific task. Neurons are present within each layer, which are the input layer, hidden layers and output layer [29].

Evaluating the SoC using a NN, involves two tasks, these tasks are evaluated for a LiFePo battery [29], but can also be used for other storage devices. These tasks are: curve fitting and estimating values [29]. Curve fitting approximates the SoC based on a known flow, and estimating values estimates the SoC for an unknown flow.

Feedforward Neural Network

Feedforward neural networks (FFNN) can model any non-linear system [30]. They are one of the simplest NN as they are matrix based and the information only moves in one direction, forward [29]. Once the network is trained, it provides fast computational speed and does not have intensive calculations that require partial differential equations. Chemali, Ephrem and Kollmeyer [30] take the average of its input variables in order to achieve faster computational times and reduce the amount of memory required. The FFNN works best for curve fitting, and its accuracy in [29] behaves really well from 100% until 60% SoC, after this the error increases exponentially.

Radial Bias Function Neural Network

More commonly, the Radial basis function (RBF) uses powerful computational tools that can solve estimation issues. The RBFNN is known to be used when the data contains incomplete information, and is able to provide a good approximation as well as performance [31]. The RBFNN receives input values of current, voltage and the desired error goal, which can be used to produce a specific number of neurons in the hidden layer [29]. It can be used to analyze the relationships between sequences. The RBFNN neural network has been used in SoC estimation and has been tested for LiFePo and lithium poly carbon monoflouride (Li/CFx) batteries [29] [32]. The results showed the speed and estimation accuracy meet the estimation demands, Enache and Diaconescu [29] found that the maximum error achieved was far below the desired error of 0.01% for a 0.55A discharge current, but yielded much higher errors when using a 0.35A discharge current. While, Linda et. al. [32] presented an average relative error of 2%. Chemali et. al [30] offer a competitive estimation performance with a mean absolute error (MAE) below 1% and Chang [33] for an RBFNN achieved an error of 0.02%.

Backward Propogation Neural Network

Backwards propagation (BP)NN is a FFNN that uses a three-layer model and is a type of artificial neural network (ANN). BPNN is involved in SoC estimation as it is able to provide nonlinear mapping and self-learning [34]. It is able to estimate the SoC by using historic values of the inputs, for example: voltage, current, and the temperature of a battery [32] [19]. This can be altered to be used for other domains. Hannan et. al. [24] have developed a robust estimation strategy for a Li-ion battery, that was able to use less parameters, less complex equations and not use a battery model, all while still working efficiently. They achieved a root mean squared error (RMSE) result of $0.0183, 0.0148, 0.0119$ at different temperatures of: $0°C$, $25°C$, and $45°C$. Zhang et. al. achieves a maximum estimation error of a BPNN in a working condition test of 4% for a li-ion battery.

Recurrent Neural Network

To contrast the NNs presented, recurrent neural networks (RNN) are a type of ANN that uses time series data, so that the output of the previous step can be used as the input for the current step [35].
RNNs are usually a feed-forward networks where the output is fed back and used as an input variable. There are three different types of RNNs; Many-to-many, One-to-many, and Many-to-one [36], as shown in Figure 1. The Many-to-many uses a 'sequence of inputs and outputs a sequence of outputs, the One-to-many takes in single inputs and outputs a sequence of outputs and the Many-to-one: takes in a sequence of inputs and outputs a single output' [35]. There are two types of RNNs that can be used for SoC estimation, Gate Recurrent Neural Network (GRU)NN and Long Short Term Memory (LSTM)NN.

Li et. al [37] uses a single-GRU network that achieves an average error of 3%, however, after 30 discharges, the average error increases to 40%. This paper proposes a method to estimate SoC in the battery degradation process. Since the data obtained from the battery models are time series data, an LSTM is a good fit as it is known for its strong performance [38]. The LSTM is able to generalize abstractions and this author further confirms that LSTMs are able to do so and by learning from data taken under a different conditions, which was able to obtain an MAE of 0.573% [38]. The paper by Chen et.al [39] uses a LSTM with an extended input and constrained output to achieve better SoC estimations, they were able to have an RMSE of 1.3% and maximum error of 3.2%. Almaita [40] uses a LSTM that is able to 'avoid long-term dependency problems' and accurately estimates SoC with an error of 0.62%. Their paper shows that LSTMs can excellently monitor and control battery systems [40].

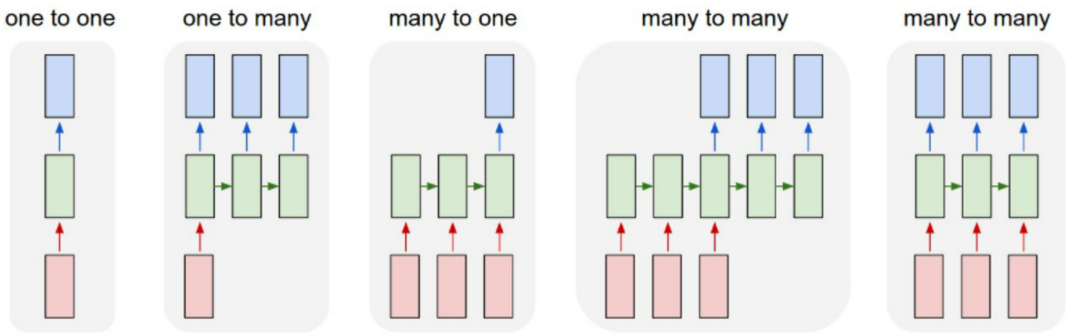


Figure 1: Types of Recurrent Neural Networks [36]

Neural Network Summary

For a neural network to yield the best result a large amount of training data is required to provide the highest quality test. Each neural network framework requires slightly different data. In [41] a FFNN is trained with voltage, current, SoC, and the temperature which was obtained by measuring the constant current discharge at different intervals.

On the other hand, the RBFNN that is presented in [33] trained with data that was collected from 10 different values of discharging current. Multiple tests were done with these varying currents to verify its performance with slightly different data, this was also compared to a FFNN. RBFNN absolute error was slightly lower than a FFNN, the average absolute error of the RBFNN was between 0.0008 and 0.002, and the FFNN was between 0.064 and 0.127 [33] [29]. Enache [29] concluded that FFNN can be used in all systems that want to estimate SoC for batteries. The RNN uses voltage, current, and temperature and the last SoC value as inputs, where the GRU network, the average error achieved is less than 3% [38], while the LSTM achieved a maximum error of 3.2% [39]. This is a contrast to Almaita [40], who achieves an error of 0.62% for the LSTM.

2.5 Selecting an Algorithm

Selecting which model to use is based on a number of factors. It is dependent on the amount and the quality of the data, as well as the quality of the results that are desired and the 'interpretability of the model' [42]. The results presented use different metrics in order to measure the accuracy of the SoC estimation. RMSE, MSE and MAE are used. They are different kinds of errors that are further explained in Section 3.4.

It is impossible to directly compare the errors of the different models presented, since the errors are relative and dependent on the data used to train the algorithm. Even if all models use RMSE, a direct comparison is not valid. There, selecting a model will not only include the error metrics used, but is based on the complexity of the model and if it can be generalized.

Table 1: Algorithm Comparison

Algorithms	Results	Ref
FFNN	The FF network discharges at different starting SoC. The maximum average error is about 3.6% and the maximum error appears at 20% SoC, with an error of 25%.	[29]
FFNN	The test procedure performs a full charge at 25 degrees Celcius, results in MAEs below 1%.	[30]
RBFNN	Discharges at different rates for different lengths of time. Uses an algorithm to select best NN feature. With a average absolute percentage error of 0.021% and a maximum absolute percentage error of 1.82%.	[33]
BPNN	Developed a robust estimation strategy, that is able to use less parameters, less complex equations. Achieved a root mean squared error (RMSE) result of 0.0183, 0.0148, 0.0119 at different temperatures of: 0C, 25C, and 45C.	[24]
RNN	The LSTM uses current, voltage temperature as an input and properly calculated the outputted SoC with a maximum standard error (MSE) of less than 0.62%	[37]
RNN	The LSTM is trained from data taken under different conditions, which is able to obtain an MAE of 0.573%	[38]
RNN	Uses a LSTM with an extended input and constrained output for SoC estimations, they were able to have an RMSE of 1.3% and maximum error of 3.2	[39]
RNN	LSTM that accurately estimates SoC with an error of 0.62%	[40]

NNs stand above the other models due to the accuracy that they can obtain with low complexity. This is clear in the estimation of SoC, where NNs from [19] clearly outperformed methods and is the chosen preferred approach. This paper is a review of SoC estimation methods for batteries. Table 1 shows a brief outline of the results of the NN presented in the literature. NNs perform well on systems that are rich in data. Therefore, a NN is chosen for the estimation of SoC for this research.

Neural networks are most suited for this research but will require a parameter sweep to find the best values for the hidden layer neurons, learning rate and other parameters that are involved [24]. A choice also will be made on the type of network that will be used. Hannan et. al. [24] used a neural network in combination of an optimization technique to find the best values of neurons for the hidden layer neurons and learning rate. Although there are many NNs to choose from, a time series NN is optimal, since SoC depends on the previous SoC in order to estimate the next SoC. Although the results are relative, Table 1 shows that the RNN has one of the lower errors and is the only NN presented that is meant for time series problems. This narrows down the NN to an LSTM.

In Summary

Although many of these algorithms produce good results, the issue again lies with the fact that they are particular to a specific kind of storage device. Further research needs to be done on hybrid methods. There is also an abundant trade off between computational efficiency and accuracy. A more generalised estimation method needs to be made that can be used to cover various types of storage devices that will also be able to yield a good result and not take too much computational effort. A LSTM will be used as it can yield an accurate results with not too much computational effort.

3 Research Methodology

In this chapter, an overview of the methodology is presented, based on the problem statement and literature review. The data that is used for each model is explained and visualised. The two models presented in this chapter are the ANN and the LSTM, their evaluation metrics and their implementation. The first model for the ANN is the test case and the LSTM is the main model used for this research.

3.1 Data set for Models

The data being used in the ANN and LSTM models of this thesis are: three different lead acid batteries (Conrad, Yuasa and Multipower), one lithium battery (LiFePo) and a hot water buffer.

The battery data provided by Homan [8] includes various types of data for various kinds of batteries, for example 'the Conrad CP672 valve regulated lead acid battery, the Yuasa NP7-6 valve regulated lead acid battery and the Multipower MP7-6S lead acid battery' [8] and a LiFePo battery.

The data that is used for the models are: charging, discharging cycles at six different rates and a realistic usage scenario, as shown in Figure 3. The realistic usage scenario is a data set where the battery discharges and charges as it would in a real world system. Figure 2 displays the current and voltage of a charging Conrad battery. The current and voltage data is present for all six charge/discharge cycles and used as the input for the ANN model.

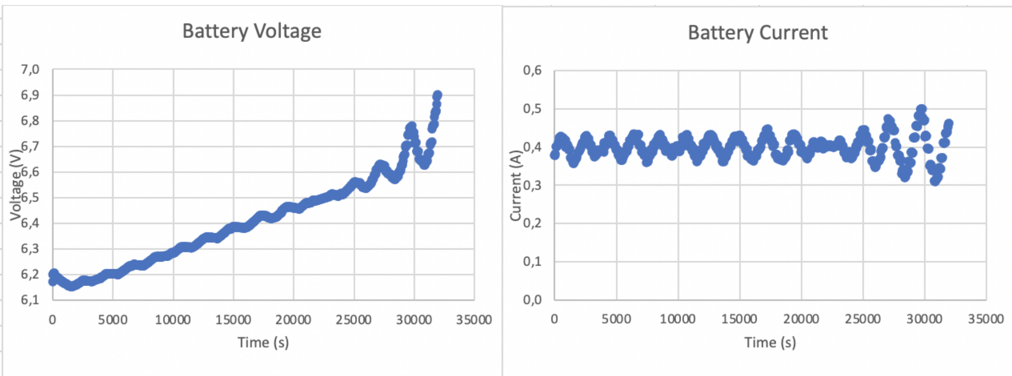


Figure 2: Voltage and Current of a Charging Conrad Battery at 400mA [8]

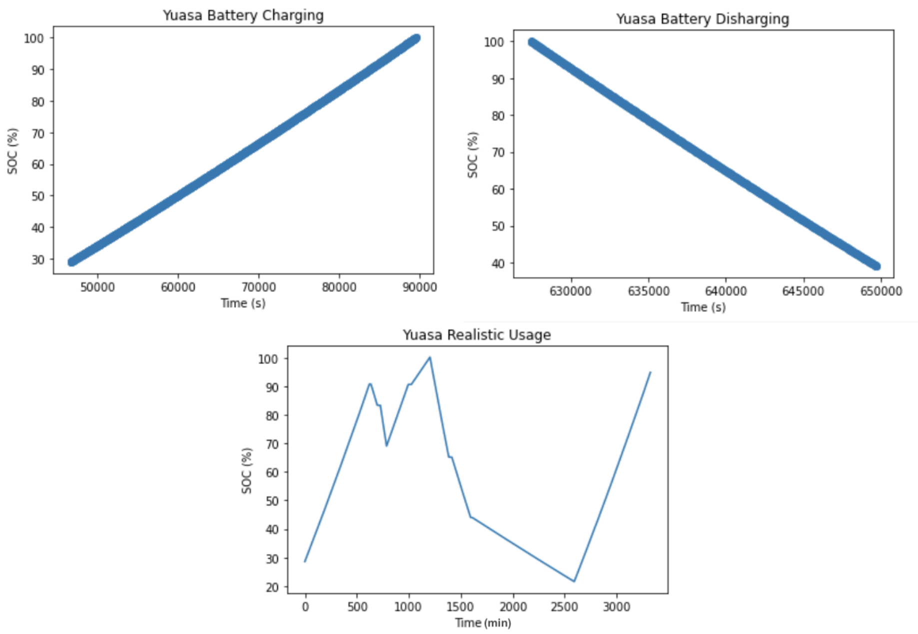


Figure 3: SoC of Charging, Discharging, Realistic Usage Battery [8]

The data used as input features to train the LSTM model is shown in Figure 4. The LSTM evaluates the model using the realistic usage scenario from Figure 3.

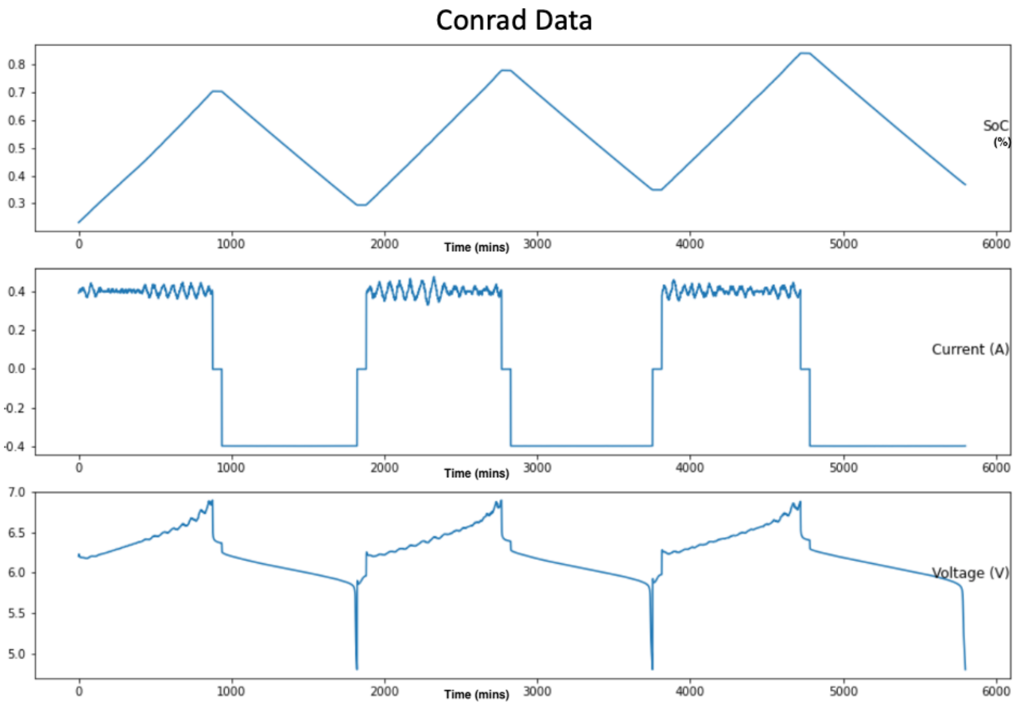


Figure 4: Visulization of the Data

Other than only using batteries, the hot water buffer data is used to test the versatility of the LSTM model presented in this thesis. The hot water buffer data is taken from van Leeuwen [43]. Such as in Figure 5 where the storage is discharged to the minimum SoC. i.e. when the water temperature reaches its lower threshold. The data that is used are the time, change in temperature, flow and SoC.

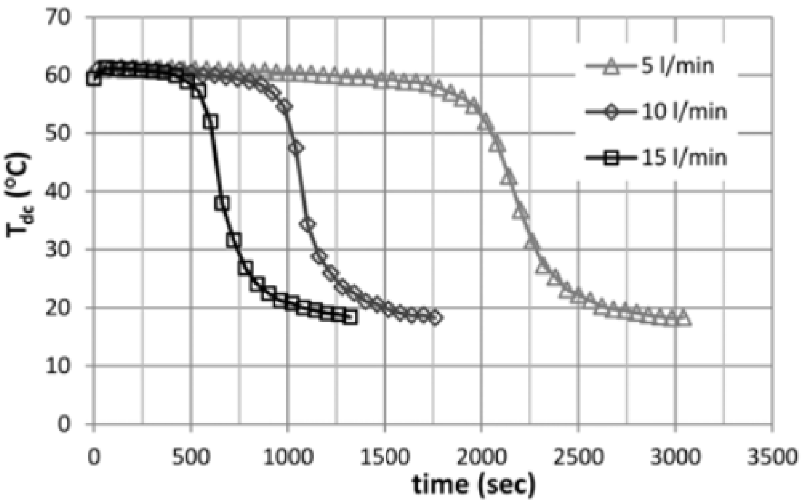


Figure 5: Water temperature at constant flow rates [43]

Data Pre-processing

Pre-processing the data is an important part of using machine learning models. The SoC is calculated using the information provided from the data set. The SoC is directly calculated for each battery, as shown in Equation (1), where SoC_t is the battery SoC at time t, SoC_{t-1} is the previous SoC. The change of the SoC is based on the current $\bar{I}$ and voltage $\bar{U}$ in their corresponding time interval [8] and is divided by the maximum energy content $E_{\max}$. $E_{\max}$ is calculated by multiplying the rated capacity (Ah) by the nominal voltage (V), these values are presented in Batteries in Smart Microgrids [8].

$$SoC_t = SoC_{t-1} + \frac{\bar{U} \cdot \bar{I} \cdot (t - (t-1))}{E_{\max}} \tag{1}$$

This is the SoC that the models will use as measured value, which will be estimated by the neural networks in Chapter 4. The features that are selected for the input and outputs vary slightly for the LSTM and ANN models. The ANN model uses the time, current and voltage as inputs and the SoC as the output, while the LTSM model uses historical SoC, current and voltage as inputs and the SoC as the output. The LTSM model relies on previous time steps as inputs, therefore the SoC_{t-1} is an input.

For the hot water buffer, the SoC from van Leeuwen [43] is calculated as shown in Equation (2). Where SoC_t charging thermal energy at time t, S_t is the stored energy and S_{max} is the maximum charged capacity.

$$SoC_t = \frac{S_t}{S_{\max}}, 0 \leq SoC_t \leq 1 \tag{2}$$

The data being used in the models also has to be scaled before being run through the neural network. This means that all the input and output features need to be scaled between a value of 0 and 1. A more detailed explanation of this is will follow in Section 3.3.

3.2 Test Plan

The test plan identifies the core plan used to test the selected models. Although the models used are different, they follow a similar testing approach. The following approach will be used for both models. The machine learning workflow works iteratively and is summarized with the following steps:

- **Define objective:** Know what the objective is as this will influence what kind of algorithm and data can be used.
- **Collect data:** Ensure quality data so the algorithm can find the correct patterns to learn from.
- **Prepare data:** Cleaning the data to get rid of unwanted data and missing values. Split the data into a training and testing set.
- **Select algorithm:** Select which ML algorithm is best suited for the issue and data set on hand.
- **Train model:** Pass the prepared data to the model to find patterns and make predictions.
- **Test model:** Use the trained model on a different data set to test its output using evaluation metrics.
- **Make an estimation:** Estimate future SoC.

The training of the models relies heavily on its data. Therefore the used ML technique is dependent on available input data. Based on the steps presented, the machine learning process can be split into two phases. Phase 1 is the learning phase, where the pre-processing, learning and testing will take place. The pre-processing step involves cleaning and formatting the data. The learning step takes the data for supervised and unsupervised learning. The final step of this phase is the testing, to make sure that the results coming out matches the data set. Phase 2 involves estimation, it uses the trained model and new data in order to make an estimation. This phase is about validation and the evaluation of the model.

3.3 Method

This section describes and explains the approach used to create the ANN and the LSTM neural networks.

Artificial Neural Network

The ANN will be used as the test case for this research, to see how a simple NN compares to a time-series NN. The artificial neural network uses multilayer perceptrons (MLP) which is a FF algorithm. The MLP one of the easiest to test, so it will be used as a baseline estimate. The MLP network can consist of three layers or more. These layers are the output and input layers with a hidden layer. Each neuron consists of a set of weights to which the respective inputs are multiplied. It applies an activation function where the output becomes the input in the next layer [44]. In this thesis, the MLP uses the battery data presented in Figures 2-3, and takes the current and the voltage and maps these inputs to the output. Figure 6 shows the MLP architecture used to determine the SoC, and is adapted from Kashkooli et. al [45].

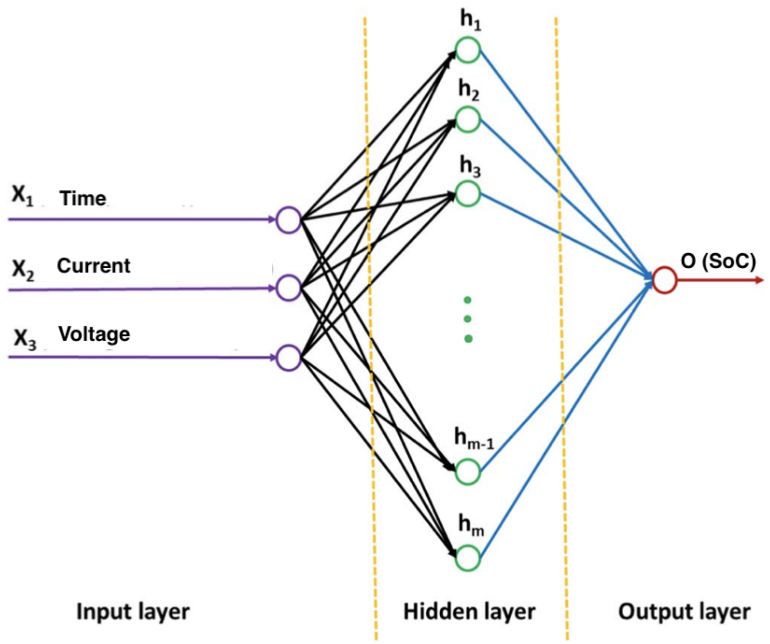


Figure 6: MLP architecture for SoC estimation [45]

The input layer has 2 neurons (excluding X_1), the hidden layer has m neurons, and the output layer O has one neuron. The nodes in the layers are related to next layer nodes which have a weight w_{im} [19] [45]. 'The total input of a hidden neuron h is computed using weights and bias of each layer' as shown in Equation (3) [19] [45]:

$$h_m = \sum_{i=1}^{n} w_{im} x_i + b_m = O \tag{3}$$

where h_m is total input of the hidden layer with m neurons. x_i is the input to the hidden layer with i neurons. b_m is the bias of the hidden layer [45].

$$relu = max(0, x) \tag{4}$$

$$\sigma = \frac{1}{1 + e^{-x}} \tag{5}$$

$$tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

Different activation functions can be used on the hidden layer, for example: *relu*, hyperbolic tangent and sigmoid, as shown in Equations (4-6). These functions select the data that are propagated through the network.
The created MLP for this research is activated with a *relu* function. *Relu* is more computationally efficient to compute than sigmoid functions and shows better convergence performance [22].

The activation function (relu) is applied to O in the output layer in Equation (7):

$$O = relu(SoC_t) \tag{7}$$

For model training, the data must be pre-processed and prepared, as mentioned in Chapter 3.1. The data is split so that the first 30% of the battery data is used for training and the next 70% will be used for testing. The model is created by only using the training data. The model is trained on every charge/discharge cycle of the Conrad, Yuasa and Multipower batteries. This trained model is tested on the realistic usage scenario data for each battery.

Recurrent Neural Network

RNNs use autoregressive forecasting which makes one estimation at a time and feeds the output back to the model. A RNN processes a sequence of vectors, one after another. By doing this, it is able to pass the values of the previous hidden state to the next state. This hidden state acts like the memory of the neural network. The memory holds information of the previous state, which is data that the network has seen before [46]. The input and output values in the NN must be in range [-1, 1], the tanh function can do this by regulating the values. This process is necessary as the vectors flowing through the network go through many transformations and math operations. This ensures that the values do not explode, which may cause other vectors to become insignificant [46].

To remove the vanishing gradient problem, paths are created, where derivatives neither vanish nor explode. RNNs, such as an LSTM, does this, as it is able to update the weights at a given time step. It either clears the state or keeps the value.

There are three different types of RNNs that can be used; Many-to-many, One-to-many, and Many-to-one. The Many-to-one model is used for this research, a visualisation can be seen in Figure 1 in Section 2.4. It is chosen as many 'inputs' lead to one 'output'. The many-to-one model is used as the data provides multiple features that contribute to the output. For this research, voltage, current and SoC_{t-1} will be used as the inputs and SoC_t will be the output.

The LSTM structure is very similar to those used in the RNN described earlier. The difference is within the LSTM cell. The LSTM cell is shown in Figure 7. There are several operations within the LSTM cell, which allow the network to keep or forget information. An LSTM cell contains the following components [47]:

- Forget Gate f_t
- Candidate layer $\hat{C}_t$
- Input Gate I_t
- Output Gate O_t
- Hidden state H_t and H_{t-1}
- Memory state C_t and C_{t-1}
- Hyperbolic tangent Activation Funtion tanh
- Sigmoid Activation Function σ
- Weight Matrix $W = W_C, W_I, W_O$
- Bias Parameters b_F, b_I, b_C and b_O

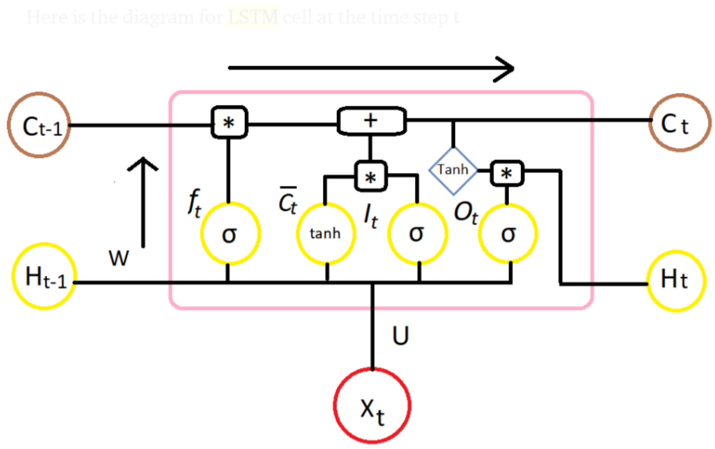


Figure 7: LSTM Cell [47]

In Figure 7, the inputs of the LSTM cell at any time are X_t, H_{t-1} and C_{t-1}. H_{t-1} and C_{t-1} are the previous Hidden and Memory state. The outputs are current Hidden and Memory state, H_t and C_t. The four yellow circles inside the pink box are Forget Gate f_t, Candidate layer $\hat{C}_t$, Input Gate I_t and Output Gate O_t, where f_t, I_t and O_t are activated with sigmoid function and $\hat{C}_t$ is activated with tanh [47] [48].

These gates concatenate H_{t-1} with the first input and apply an activation function. Vectors are created from the gates, with values between 0 and 1 for sigmoid and -1 to 1 for tanh, thus resulting in the four vectors f, C, I, O for every time step [46]. The LSTM gates protect and control the cell state, it is a way to let information through.

The first step to calculating C_t uses the previous state C_{t-1} and multiplies it with the forget gate f as seen in Equation (8) [46]. It then multiplies I_t by $\hat{C}_t$ as shown Equation (12).

$$C_t = C_{t-1} \cdot f_t \tag{8}$$

The forget gate is formulaized in Equation (9) [46]:

$$f_t = \tanh\left(W_C \cdot [H_{t-1}, X_t] + b_F\right) \tag{9}$$

When the value of f is 0, the previous memory state will then be forgotten, but when the value is 1, the memory will be passed on. The forget gate F is just adding the two inputs with weights. The new memory state can be calculated from the input state I_t and C_t.

An LSTM has a chain like structure as can be seen in Figure 8 below. It contains four interacting layers, as opposed to one. In this figure there are pink circles and yellow boxes. The pink represents pointwise operations, like vector addition. The yellow represents trained neural network layers [49].

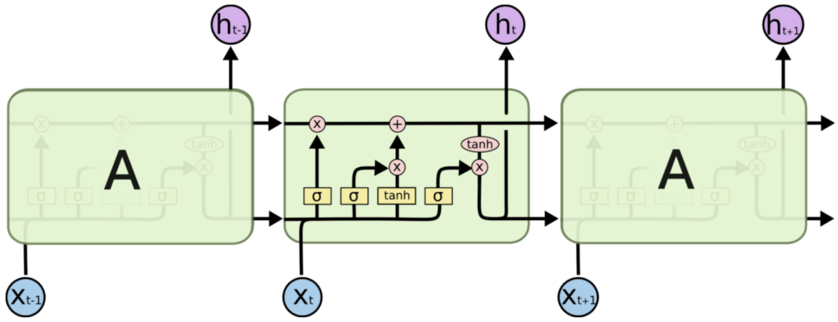


Figure 8: LSTM Cell Chain Structure [49]

The step-by-step approach to an LSTM is presented in the equations below. The LSTM looks at $[H_{t-1}, X_t]$, where the output of C_{t-1} is a value between [0,1]. Where 1 keeps the memory and at 0 forgets it.
The next step has two parts, as shown in Equation (10) and (11) [46] [48]. The sigmoid layer σ decides which values get updates. The tanh layer creates vectors of C_t, that are added to the state [50]. b_I and b_C are different the biases. The bias is added to each layer and acts as an offset.

$$I_t = \sigma\left(W_I \cdot [H_{t-1}, X_t] + b_I\right) \tag{10}$$
$$\hat{C}_t = \tanh\left(W_C \cdot [H_{t-1}, X_t] + b_C\right) \tag{11}$$

Equation (12) [46] shows how the previous cell state C_{t-1} can be updated. The new state C_t is calculated by multiplying the previous state by F_t, and by adding $I_t \cdot \hat{C}_t$.

$$C_t = f_t \cdot C_{t-1} + I_t \cdot \hat{C}_t \tag{12}$$

The output is a filtered version of the cell state. The sigmoid layer decides what the output will be based, as shown in Equation (13) [46]. The cell state is then outputted through tanh and multiplied by the sigmoids gate output shown in Equation (14) [46].

$$O_t = \sigma\left(W_O \cdot [H_{t-1}, X_t] + b_O\right) \tag{13}$$
$$H_t = O_t \cdot \tanh\left(C_t\right) \tag{14}$$

The model uses 3 inputs, SoC_{t-1} the current and voltage and outputs the SoC_t that gets fed back into the model as shown in Figure 9. Figure 9 has been adapted and shows how the LSTM cell fits into the network, T_t as shown in the figure can be ignored as it is not used as an input. This model is used with four different batteries and a hot water buffer. For the hot water buffer, the volume flow (L/s) relates to the current (A) and the temperature $(°C)$ relates to the voltage (V).

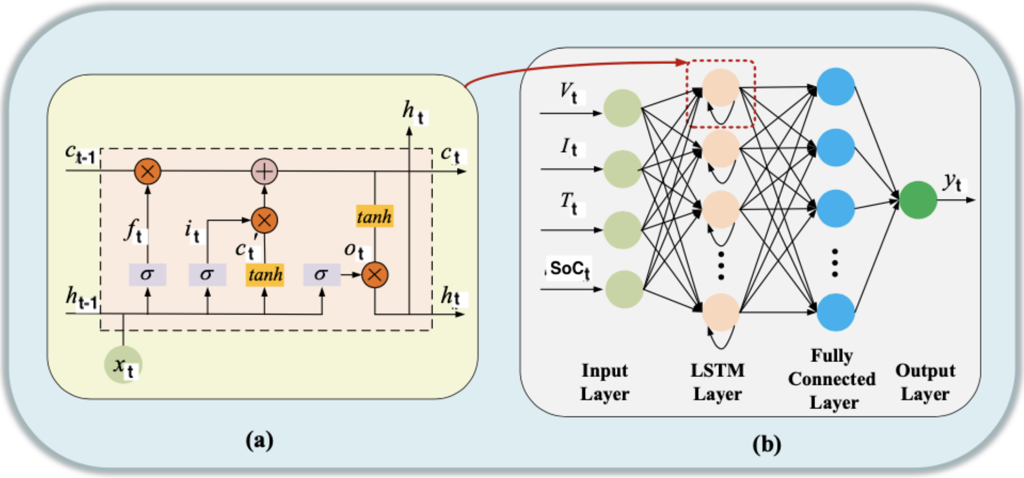


Figure 9: Adapted LSTM topology. (a) Structure of LSTM unit; (b) Structure of LSTM network [48]

Figure 9 shows the LSTM with one hidden later, the model input includes voltage, current and SoC: $x_t = [V_k, I_t, SoC_{t-1}]$. The SoC input here is the previous SoC value so SoC_{t-1}. The estimated SoC_t can be attained in Equation (15) [48]:

$$\begin{cases} y_t = f\left(W_g h_t + b_g\right) \\ SoC_t = f\left(W_y g_t + b_y\right) \end{cases} \tag{15}$$

where g_t is the output of the full layer, f refers to the linear activation functions, W_g is the weight matrix, b_g is the bias parameter of the whole network, W_y is the weight matrix, b_y is the bias parameter of the output layer [48]. The MSE is used as the loss function while training the model [48].

The required input for the LSTM model is 3-dimensional. Data has to be transformed from 2-dimensions to 3-dimensions. The new dimensions are [samples, timesteps, features]. An LSTM has a lookback value, which is how many previous input steps the model should consider when updating its weights [48]. A parameter test is done to find the value of the lookback that outputs the best results.

After the data transformation is done for the target variable, the data set is divided, where the first 30% of it is used for testing and the next 70% of it will be used for training. A further explanation of the implementation is given in Section 3.5.

3.4 Evaluation Approach

The final output of the ANN and LSTM must be validated and evaluated. This means that the trained data model will be evaluated with test data [51]. The ANN with be evaluated only with the metrics displayed in Equations (16-18), and the LSTM will be additionally evaluated by retraining the model and then applying these metrics again. As part of the research goal, it must be investigated how the accuracy of the model can be quantified. There are other metrics availble when dealing with NN. Those metrics, for example, are: Area Under Curve and F1 Score deal with different problems and are not relevent for this research.

An issue with estimating the SoC using machine learning is that the model at a certain instance in time can yield a good result, but over time the errors can accumulate and increase. This means that, a model can yield a good result in one metric and a poor result in another. As mentioned in the literature, the metrics cannot be directly compared as they are dependent on the data used for training.

The mean absolute error (MAE) and the mean squared error (MSE) are used to evaluate the model [52]. MAE is the average of the difference between the original values and the estimated values [52]. It is a reflection of the difference between the estimations and the actual output. The MSE is the average of squared differences between the estimated output and the true output [52]. The root mean squared error (RMSE) measures the average magnitude of the error [52].

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |(y_t - \hat{y}_t)| \tag{16}$$

$$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \tag{17}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \tag{18}$$

Where: y_i is the estimation, $\hat{y}_i$ is the true value and n is the total number of data points.

MAE, MSE and RMSE express the average model estimation error [52]. The values of these metrics are all expressed in the same unit as the target value, for this case all units will be in percentages (%).

The output values range from [0,inf). The results of these outputs favor a lower score, the lower the score, the better. Based on the equations the RMSE can only be larger or equal to the MAE. When there is a larger difference between the RMSE and MAE, it shows that there is a larger variance of errors in the data [52]. When the RMSE and the MAE are equal, it assumes that the error are the same magnitude [52].

The RMSE is taken by taking the square root of the MSE. The implication of the RMSE score is that it is more useful with larger errors. The errors $y_i - \hat{y}_i$ are squared before they are averaged, this means that the errors provide a higher weight to larger errors. The MSE can also magnify the errors if the model makes a single bad prediction. A disadvantage of the MAE is that if there are large errors coming from outliers, they will be weighted the same as lower errors, which will result in a model that can make poor predictions.

All three metrics shown in Equations (16-18) are used [52]. With all these metrics, there is not a good value as they cannot be judged absolutely and can only be judged comparatively. Such as, RMSE penalizes large errors, which can make MAE more appropriate at times, but RMSE and MAE should be as low as possible [53].

The final metric is the accuracy metric, as shown in Equation (19) [52]. It estimates the difference between two sets of results, the estimated value and the true value. For example, given two lists, $\hat{y}$ and y_i, for every position index i, compare the i-th element of $\hat{y}$ with the i-th element of y_i and perform the following calculation: Count the number of matches and divide it by the number of samples. This value goes from [0,1], with 1 being 100% accuracy.

$$\text{accuracy} = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t) \tag{19}$$

The accuracy will be the main metric that will be used to compare the model. It is able to provide the value of how off the model is from the measured data.

For the LSTM, the next step to evaluating the model is to retrain the model using a different data set. The model is retrained with the Yuasa and LiFePo data. The model outputs are re-evaluted using the same metrics presented in this section and compared to the initial model metrics. The purpose of this is to find out how the model performs with different batteries and data sources.

3.5 Implementation of Machine Learning Models

Python is a commonly used programming language amongst data scientists as there are many libraries that can implement various data related functionality. From the Python libraries, Keras and Tensorflow frameworks can easily be used to prototype and test ML architectures rapidly and offer multiple levels of abstraction [38]. The hyper-parameters for the neural networks are defined in this section.

Artificial Neural Network

The algorithm uses current and voltage as an input with the SoC being the output. Using a machine learning library the network will be trained by using the different charging and discharging experiments from Homan for training [8]. The first step is to train the ANN based on all the charging and discharging data and then validate this model with the realistic usage data set.

The ANN has 100 neurons in the hidden layer and is activated with *relu*, with a learning rate of 0.001. The model is fitted for 500 training epochs with a batch size of 200. Due to the smaller data size, one hidden layer is used. The number of neurons in the hidden layer can range from 10-100 neurons. 100 neuron are used to predict more complex systems and 10 neurons for less. 100 neurons is taken to get the most accurate result.

The choice of activation function is explained in Section 3.4. The learning rate controls the step size for the model. Using a high learning rate can result in a faster model. The drawback is that it may miss the minimum loss function. The opposite happens when selecting a lower learning rate, the lower the learning the rate the more likely it is to find the minimum loss function. There is a trade off between a high versus low learning rate. The trade off is that lowering the learning rate will require higher epochs. Therefore, 0.001 is chosen as the learning rate and 500 is chosen for the number of epochs. The batch size is the number of training data sub-samples for the input.

After training from the charging and discharging situations, the model is validated with the realistic usage data. The output of the different batteries are be compared and analyzed in Section 4.1. The model is fitted with the charge and discharge data set, where the first 30% of the data is used for training and the next 70% for testing. For the validation the model used the realistic usage data set to see if the model was able to learn from the charge and discharge data. The different batteries should yield similar results as they are using the same model. A variation in the results will be due to the differences in the input data.

Recurrent Neural Network

There are three important steps to be able to model an LSTM. The first step is to transform the data into something that is applicable for time-series forecasting. The second step is to prepare the data such that it can be used in an LSTM for a multivariate problem. The last step is to make an estimation and forecast the result, by transforming the data back into its original state. As part of the data preparation step, the data has to be framed as a supervised learning problem and the input features have to be normalized. By using the data of the SoC, current, voltage and time, a time-series forecasting problem can be framed. As mentioned in the previous section, the model input includes voltage, current and SoC: $x_t = [V_t, I_t, SoC_{t-1}]$, and outputs the SoC_t. Where, given the conditions of the inputs, the SoC can be forecasted in the next time step.

The model is fitted with the Conrad battery. The Conrad battery has the largest data set compared to the other batteries. The model is fitted with the Conrad data set, where the first 30% of the data is used for training and 70% for testing. The inputs are to be reshaped to be 3 dimensional, which is a format expected by the LSTM model. As mentioned in Section 3.3, the format is: [samples, time-steps, features].

The model is initialized to have sequential attributes with a learning rate of 0.001, as the lower learning rate is more likely to find the minimum loss function. The first layer adds LSTM cells as nodes. With 50 neurons in the hidden layer and 1 neuron in the output later for the estimation. With a dropout of 0.2 between the two layers. The dropout aids in not overfitting the data, and can be a value between 0 and 1. A drop out rate of 0.2 is able act as a regularizer to find the optimum bias-variance, and can help prevent overfitting. The first layer specifies what shape the data needs to be in. Adam is the optimizer, which is the combination of two optimizers, Stochastic Gradient Descent and RMSrop. MSE is the loss function and the MAE is the metric. MSE is used as the loss function when the target variable is continuous. Initially, the model is fitted for 50 epochs and has a batch size of 80. The model is tested with 300 epochs to find out which value gives the lower loss function. Due to the low learning rate, higher epochs are be needed.

Running the data with 50 epochs, provided results as seen on the right side of Figure 10. The plot on the left side of Figure 10 is run with 300 epochs. The plot with 300 epochs proves to be the better result as the measured and estimated values are closer in relation to each other. The model needs to be trained for the optimal amount of epochs, otherwise overfitting can occur. When there are too many epochs, the model learns patterns that are specific to that data set, which means it will do worse with a new data set. This will allow the model to provide a high accuracy on the training set but will fail with the test set.

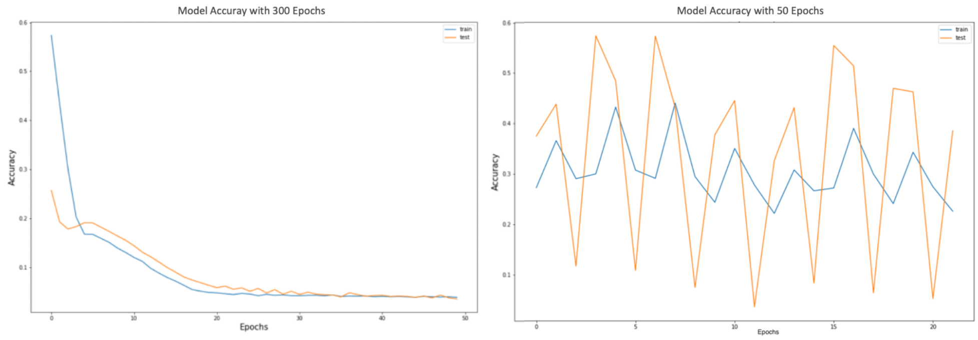


Figure 10: 300 vs 50 Epochs

In Figure 11 it is shown that the model has been overfitted on the left side, and on the right the estimation values are much closer to the true values. This correlates to the plots from Figure 10, where better outcomes were shown when the epochs were higher.

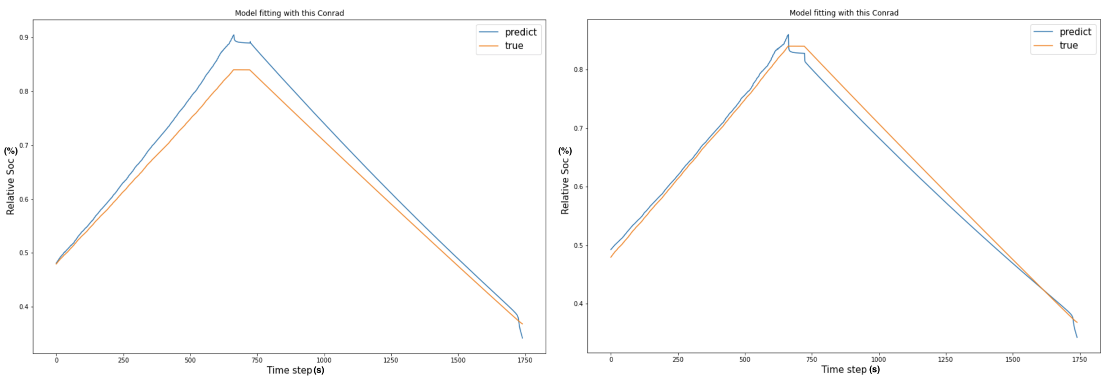


Figure 11: Overfitting the data

The model is tested with various look back values, to find the best value. A test is performed to see how the model will function with a higher lookback value. The lookback values that are trained and tested are: 5, 10 and 30. All performed worse than using a lookback of 1. These results are shown in Figures 12, 13 and 14. It can be seen that the estimation value moves further away from the true values, the higher the lookback value is. By taking more values in the lookback, the model output is reliant on these values, which is not useful when the SoC goes up or down. Therefore, only taking one value in the lookback is the best option.

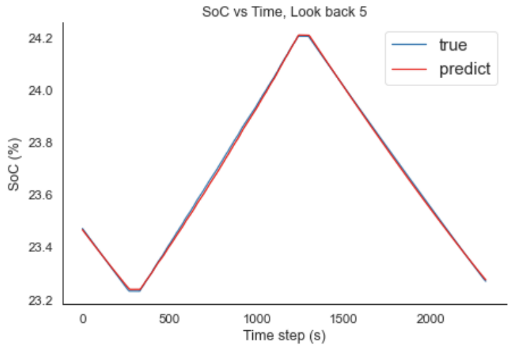


Figure 12: Conrad Results, Lookback 5

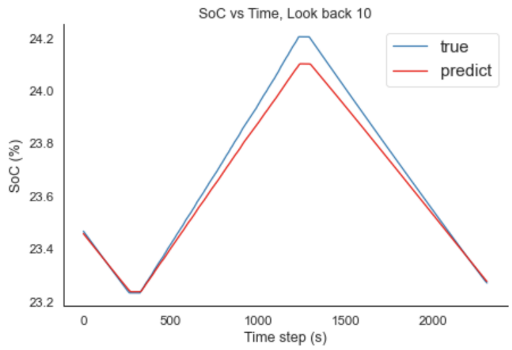


Figure 13: Conrad Results, Lookback 10

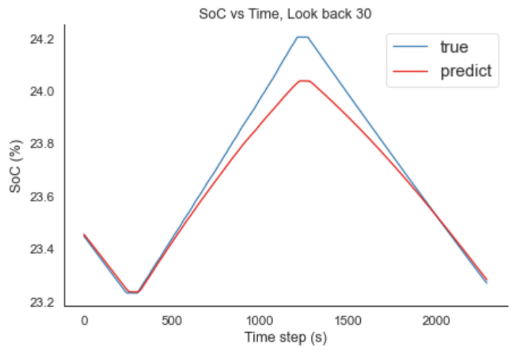


Figure 14: Conrad Results, Lookback 30

With the model being fitted with the Conrad data as shown in Figure 4, it is validated with a the realistic usage Conrad data set (Figure 3). The same model is also validated with the Yuasa and Multipower, LiFePo battery and hot water buffer to see how one battery model can be used to estimate the SoC of other storage devices.

As one of the final steps, the whole model is retrained using another battery that yields the highest accuracy estimation, which is the Yuasa. The retrained model is used to test the batteries and the hot water buffer and see how it compares to the Conrad fitted model. It is also retrained using the LiFePo, to see how the model being trained with a battery of a different chemistry will compare. The assumption is that the retrained models will both perform worse than the Conrad trained model, as ultimately they are being trained with less data. More data is likely to improve the outcome of the models. The Yuasa is assumed to outperform the LiFePo as the model is tuned for the Conrad battery and the Yuasa and Conrad are of the same battery type.

In the end, the model is able to forecast future predictions based on a known voltage and current, it does this by using the SoC output and feeding it back into the model, without the help of knowing the target values. Instead of the model estimating the whole duration of the run, it will now only predict 1 step into the future and run in a loop. In that way, it mimics the SoC estimation of a real-time system, where only the current, voltage and SoC_{t-1} is known. The same model that is used for all the tests is also used for this test, and is validated using the realistic scenario of the Conrad battery.

3.6 Testing Summary

A brief summary of the method is shown below and its results are shown in the next chapter.
ANN:

1. Test with model trained with Conrad charge and discharge data and validate with realistic usage scenario (Figure 3) for the Conrad, Yuasa and Multipower batteries.

LSTM:

1. Test with model trained with Conrad battery (Figure 4) and tested on a hot water buffer, Conrad, Yuasa, Mulitpower and Yuasa batteries.
2. Test with model trained with Yuasa and tested on Conrad, Multipower, LiFePo batteries and hot water buffer.
3. Test with model trained with LiFePo and testes on the Multipower and Yuasa batteries and hot water buffer.
4. Test with real-time SoC estimation using Figure 4 for training and Figure 3 (Conrad) realistic usage for testing.

4 Results

The following section presents the results of implementing and testing the models. Two different types of Neural Networks have been tested. A MLP ANN and a RNN LSTM.

4.1 Artificial Neural Network

The model is trained using the charge and discharge cycles for each specific battery and then validated with their realistic usage scenario as explained in the methodology in Chapter 3. This section presents the results from the Conrad, Yuasa and Multipower batteries, it does this by showing a plot of the measured values compared to the model output. The accuracy is calculated as shown in Section 3.4. The summary of the results for the ANN is presented in Table 1.

Table 1: ANN Results

Battery	RMSE (%)	MAE (%)	MSE (%)	Accuracy (%)
Conrad	9.12	7.91	83.18	25.7
Yuasa	6.59	5.77	43.40	92.1
Multipower	8.18	5.33	67.1	83.3

In Figure 15 the results of the Yuasa battery are shown. Where the orange line is the actual value and the blue line is the estimated value. The ANN model yielded an accuracy of 92%. The plot on top shows the relationship of the estimated and measured values. The model output and the measured value are plotted against each other, an ideal outcome would be for the orange line for the blue line to be as close as possible. The bottom plot shows the output of the model (blue line) compared to the actual SoC result (orange line).

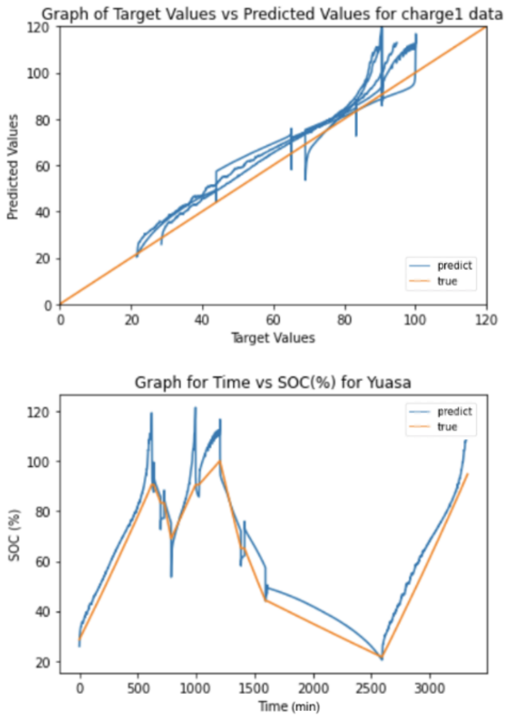


Figure 15: Yuasa Results, Estimated vs Measured Values

In Figure 16 the results of the Conrad battery are shown. Where the orange line is the actual value and the blue line is the estimated value. The ANN model yielded an accuracy of 25%. There is a clear offset in the estimated values of approximately 30%. The shape of the estimated values is similar to the shape of the real values. However, there is deviation in the peak height. The Yuasa and Conrad are implemented in the same way, but Figure 16 shows an obvious offset in the results. A reason in the differences in the results may lie in the input data and initialization weights for the Conrad battery.

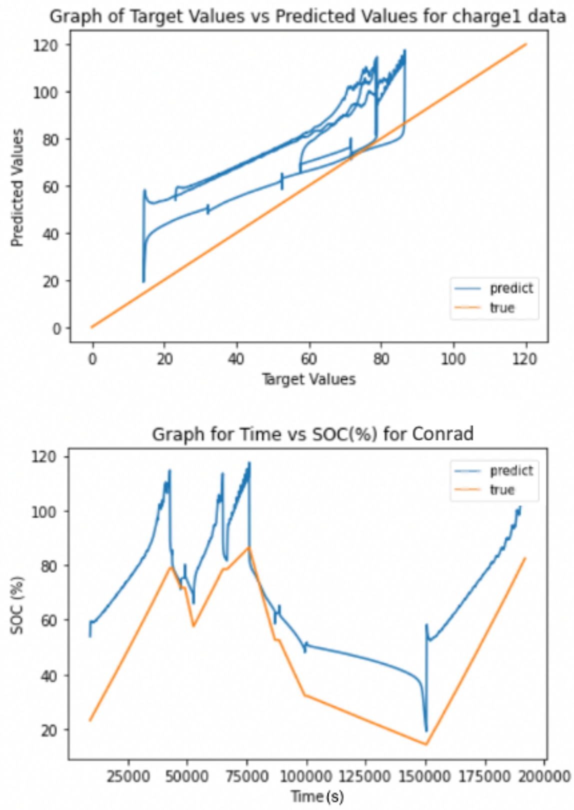


Figure 16: Conrad Results, Estimated vs Measured Values

Figure 17 the results of the Multipower battery are shown. Where the orange line is the actual value and the blue line is the estimated value. The ANN model yielded an accuracy of 83%. The estimated values follow the trend of the measured values accurately, but just like the previous 2 batteries, the peaks at around 100% are very high.

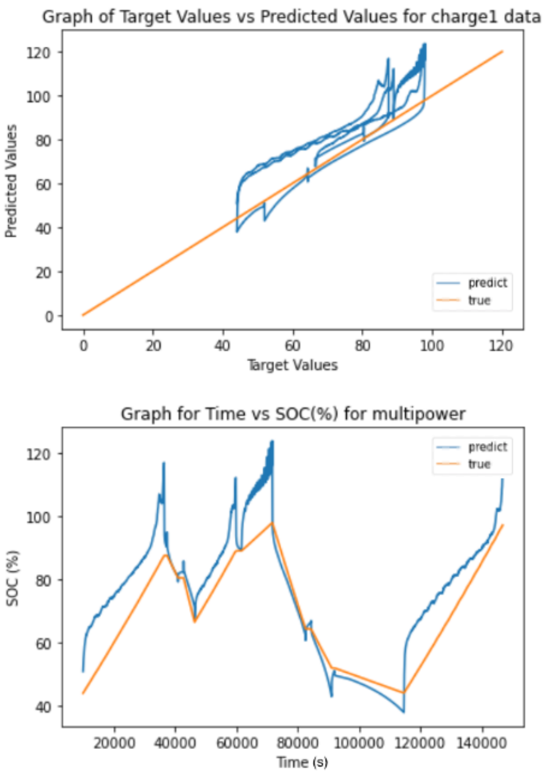


Figure 17: Multipower Results, Estimated vs Measured Values

4.2 LSTM

The summary of the LSTM results is presented in Table 2 shown below. The rest of this section provides plots to give insight to the summarized metrics in Table 2. More batteries and a hot water buffer are tested on the LSTM model compared to the ANN. As the LSTM is more complex and accurate. A larger amount of tests are required to verify whether the model is able to be generalized and used on multiple types of storage devices.

Table 2: LSTM Results, Trained with Fitted Conrad

Battery	RMSE (%)	MAE (%)	MSE (%)	Accuracy (%)
Fitted Conrad	0.010	0.013	0.000	99.8
Test Conrad	0.036	0.031	0.001	97.7
Yuasa	0.049	0.041	0.002	96.8
Multipower	2.37	2.075	6.346	96.4
LiFePo	2.69	2.511	7.265	94.4
Hot Water Buffer	0.010	0.009	0.000	92.5

The results below show that the model is validated using the Multipower, Test Conrad and Yuasa realistic usage scenarios. Figure 18 shows a slight variation in estimated SoC, where the biggest differences are seen when the SoC changes direction. The mean squared error for all the tested batteries is close to zero for all, except for the Mulitpower and LiFePo, even though the model training was only done with the fitted Conrad data, as shown in Table 2.

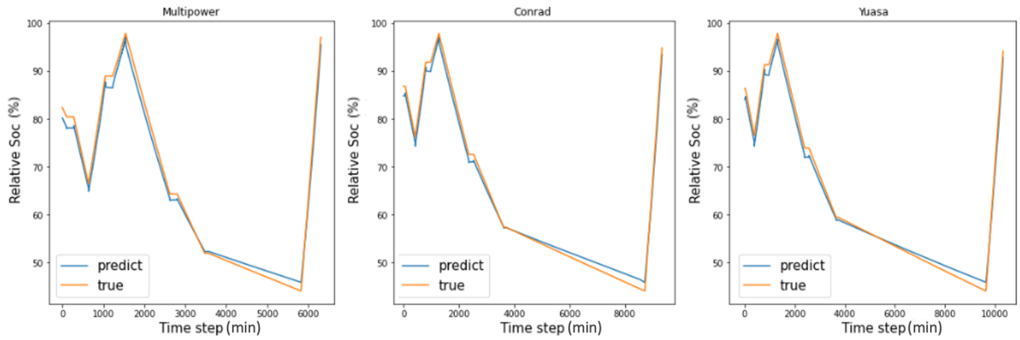


Figure 18: LSTM Results

The next part of the testing is to test a battery that is not of the same chemistry and on a different energy storage device, using same model. The Yuasa, Conrad and Multipower are all lead acid batteries. The results of the LiFePo battery is shown in Figure 19 below. This is the fourth battery used to test the model and has a different chemistry to the previously modelled batteries. The LiFePo proves to achieve quite accurate results, yielding a 94.4% accuracy with the LSTM model. The deviations from the measured values are also seen when the SoC changes direction. It has an MSE of 7.2%, which is higher than the Yuasa and Conrad results.

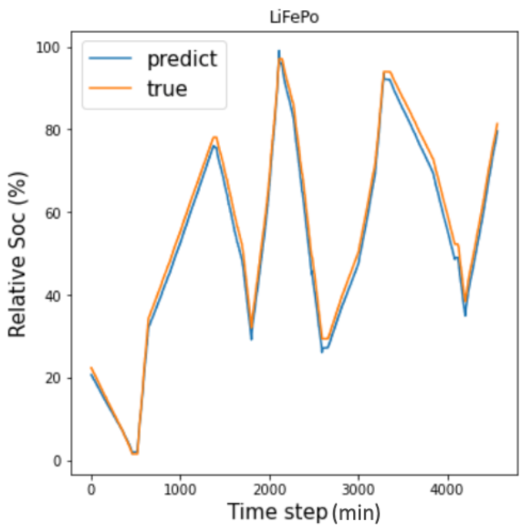


Figure 19: LSTM Results, LiFePo

Figure 20 shows the results obtained from the hot water buffer data, where it can be seen that the estimated values match the measured values very closely as it yielded a result of 92.5%. Towards 1200s when the hot water buffer is almost fully discharged, the model deviates more from the measure values.

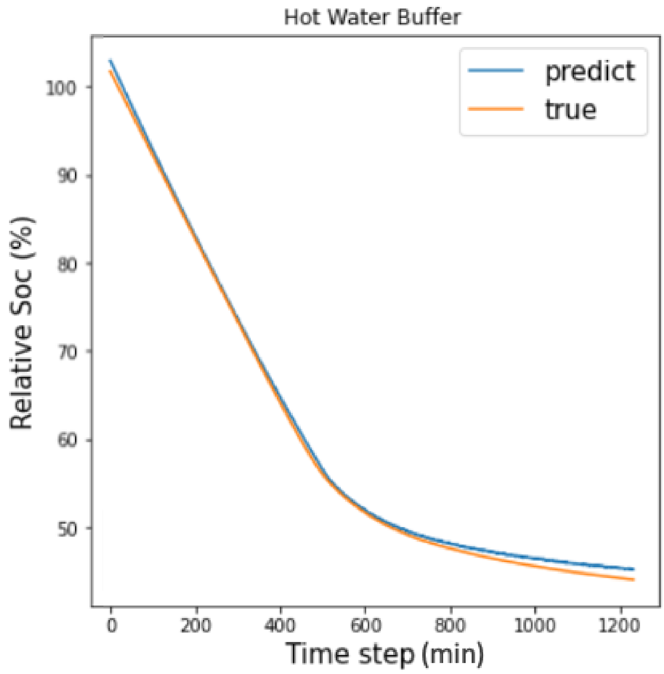


Figure 20: LSTM Results, Hot Water Buffer

Model Retraining Experiment

The LSTM model is retrained using the Yuasa data set and applied to the Conrad, Multipower, LifePo and hot water buffer data. The results of this are shown in Figure 21 and in Table 3. Afterwards, the model is retrained using the LiFePo battery, as shown by the results in Table 4, to show how the model reacts when being fitted with a battery of a different chemistry. While the initial LSTM results in Table 2 all has an accuracy of over 92.5% the Yuasa model had an accuracy range of 69.3-93.2% and the LiFePo model had an accuracy range of 84.2-96.6%

Table 3: LSTM Retrained with Yuasa Data

Battery	RMSE (%)	MAE (%)	MSE (%)	Accuracy (%)
Test Conrad	0.083	0.076	0.008	84.8
Multipower	2.13	1.95	4.573	93.2
LiFePo	7.044	5.722	49.620	81.6
Hot Water Buffer	0.124	0.0101	0.015	69.3

Table 4: LSTM Retrained with LiFePo Data

Battery	RMSE (%)	MAE (%)	MSE (%)	Accuracy (%)
Multipower	2.236	0.958	1.529	95.2
Yuasa	0.023	0.016	0.001	96.6
Hot Water Buffer	0.055	0.039	0.003	84.2

The Yuasa battery is chosen to retrain the model as it achieved the highest accuracy results for the LSTM, behind the Conrad battery. Using a different battery will yield a similar result as the accuracy shown in Table 2 is all quite similar. The accuracy is slightly lower than the first LSTM model by around 10%. More spikes are apparent at the peaks of the Multipower, Conrad and LiFePo plots. The hot water buffer values deviate from the measured values and all values for the MAE are higher. The RMSE of the Multipower battery is higher in the retrained model and is able to have an accuracy of over 90%.

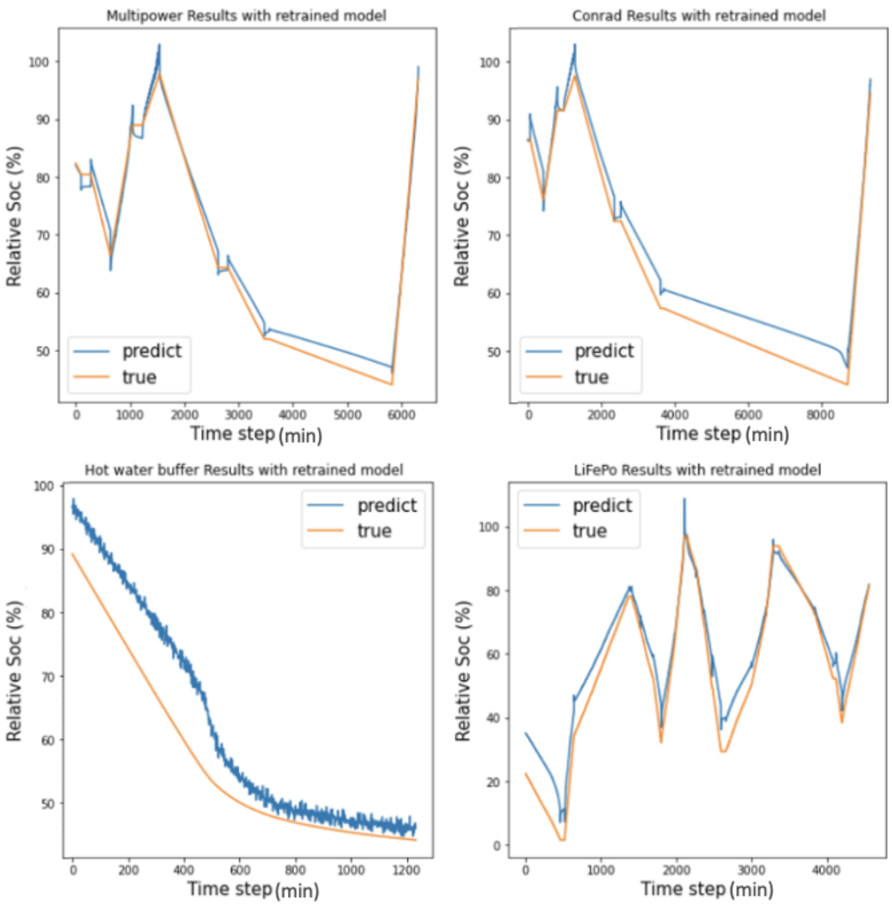


Figure 21: LSTM Results, Retrained with Yuasa model

After retraining the model using the Yuasa battery, the model is retrained using the LiFePo data, as it is a battery that has a different internal chemistry. The results are shown in Figure 22. It is shown that the retrained model with the LiFePo data performs better than that of the Yuasa retrained model. There are less deviations between the measured and estimated values and overall the accuracy is higher, when comparing Table 3 to Table 4. The lowest accuracy presented in Table 3 is 69.3%, while the lowest accuracy in Table 4 is 84.2% for the hot water buffer.

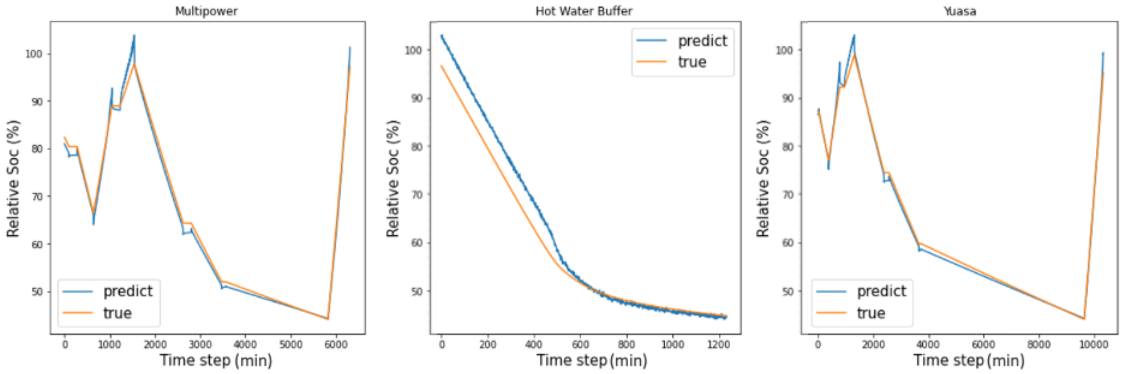


Figure 22: LSTM Results, Retrained with LiFePo model

Forecasting the LSTM Model in Real-Time

The final test is to see if the model performs well when forecasting is applied. As shown in Figure 23, the forecast does not follow the shape of the measured values as closely as the other results. As it uses the the known SoC, voltage and current and is fed the next step of the voltage and current in order to forecast the next SoC. The shapes of the values are very similar to the input shape of the voltage.

Table 5: LSTM Real-Time Forecast

Battery	RMSE (%)	MAE (%)	MSE (%)	Accuracy (%)
Conrad	19.714	14.469	388.643	70.1

The resulting accuracy of the forecast is 70.1%, and has a high MAE, MSE and RMSE due to the lower accuracy compared to the other results presented. The results in Figure 23 follows the trend of the measured results, there is just large differences when the battery is idle, for example at 3500s, the SoC drops significantly.

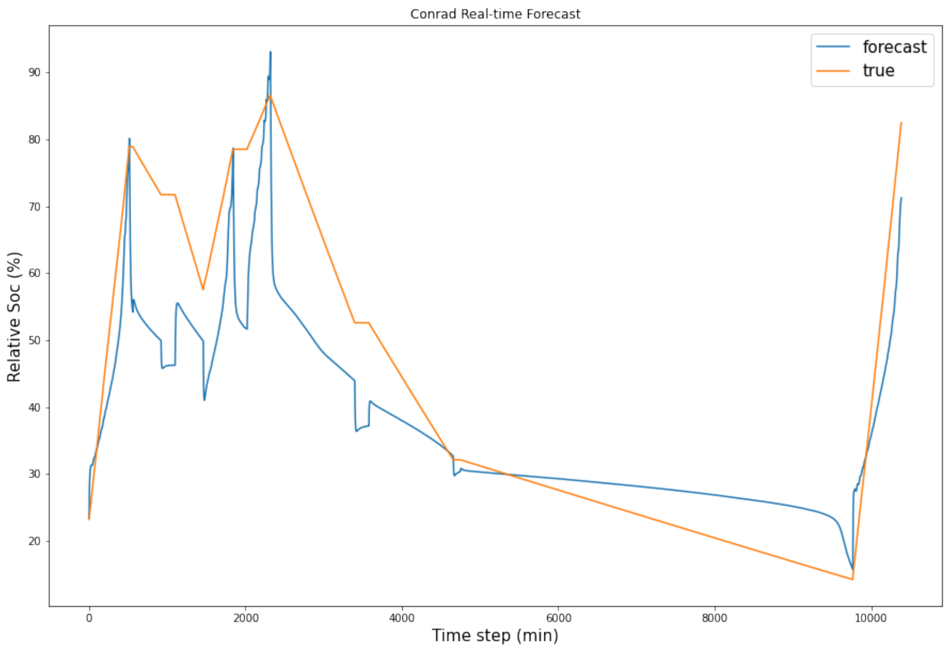


Figure 23: LSTM Results, Forecasted in Real-Time with Conrad

4.3 In Summary

The LSTM is able to estimate the SoC more accurately than the ANN model. With the ANN achieving an average accuracy of 67% across all batteries where the Yuasa battery performed the best with this model with a 91% accuracy, with the Multipower battery performing with 83.8% and the Conrad battery having the lowest accuracy of 25.5%.

The highest accuracy for the LSTM is 99.8% and 97.7% for the fitted Conrad and the Test Conrad, the other batteries are not far behind, as shown by Table 2. The model is able to achieve an accuracy of over 90% for all batteries. Compared to the ANN model, where only the Yuasa had an accuracy of over 90%. When retraining the model with the Yuasa data set, it is noticeable that the model is less accurate then the Conrad data set, as predicted. However, this is not the case for the Multipower battery, as it performs just as well as the Conrad data set. When retraining with the LiFePo data, the results are similar to the results of the initial model, even though they have different chemistries.

Forecasting the model is proven to be much less accurate, as shown in Figure 23, being the second least accurate result with 70.1% accuracy. The only result that is worse is the hot water buffer results after retraining the model, as shown in Table 3. More insight into the intricacies of the results are discussed in Chapter 5.

5 Discussion

The discussion is divided into 2 parts. Section 5.1 discusses the results from the ANN, the the LSTM and compares the results from both models. Section 5.2 evaluates the data that is used for the models. The purpose of this research is to see how well machine learning models can estimate state of charge.

5.1 Models

Two model are used in this research, one simple neural network and one time series neural network. During the pre-processing of the data, features are selected. There are properties that can be directly linked to the SoC, such as the current and voltage. The choice is made to select the Current, Voltage and SoC, as they directly contribute to SoC estimation. These three features keep the model simple so that it can be generalized.

ANN

The ANN performed accurately in estimating two out of the three batteries SoC values, which are the Yuasa and the Multipower. All three batteries are trained and modeled as explained in Section 3.3. The only differences are in the data. These differences can be due to the differences in collecting/measuring the battery data. Differences in data quality can cause the model to be more or less accurate with its estimations.

Figure 15-17 show the results for the ANN model, where the Yuasa yielded the highest accuracy of 92%, whilst the Conrad battery yielded the worst result at 25%. The results from the Conrad battery are the only results where the values are at an offset, the estimated values follow the trend of the measured data but SoC is 40% higher, where the estimated value start at 60% SoC and the actual value starts at around 20% SoC. The Conrad battery may have had an initialization problem that caused this offset. These parameters include the weights and bias of the network. These values are initially randomly initialized, the model then learns from it. Initializing the weights too small or too large can cause issues. To find the appropriate value, the mean should be zero and the variance should be the same on each layer for the activation function. Figure 15 and 17 show much better results. The visible trend between all of the results is that there is an exponential increase in the estimated data (blue line) at the peaks. This follows the trend of the input voltage..

The output of all the batteries showed high peaks when the battery is charging, and causes the estimated SoC to go beyond 100%, as the peaks follow the shape of the input voltage as shown in a similar fashion in Figure 4. This shows that the model is heavily influenced by the voltage data. To increase the accuracy, the voltage data shown in Figure 2 can be filtered so that no more oscillations occur. This should be able to reduce the peaks in the results. Clipping the SoC output at 100%, would result in a better accuracy, but influence of the voltage would be less noticeable. Another key fact is that the model is only trained using charging and discharging cycles and not with a usage scenario, since it had to be validated with the usage scenario. This can explain the peaks shown in the outputs. As shown in the voltage of Figure 2 and the SoC of Figure 3, when the battery is almost fully charged the voltage exponentially increases. Which causes the results shown in Figures 15-17 to have high peaks when reaching 100% SoC.

LSTM

The LSTM deals with a time series data problem, where previous and current values can effect the future output. LSTMs are powerful tools for time series estimation problems. One key observation is, if there is enough data for an accurate estimation.

When dealing with the lookback values from the tests shown in Figures 12-14. Saud and Shakya [54] concluded that a suitable lookback value of less than 5 is optimal and higher values will result in a poor estimation. Therefore a lookback of 1 is chosen to proceed with the model.

The LSTM achieves good results, as the obtained accuracy for all tests are above 90%, when the model is initially trained with the Conrad data-set. The model is trained and fitted with the data-set shown in Figure 6 and tested with the same data that the ANN uses for testing. The LSTM model that is created is a generic LSTM and uses 3 inputs and 1 output. This generic model achieved accurate results when tested on the Multipower and Yuasa battery. The reasoning for the accurate results could be due to the similar battery chemistries. The model takes the relationship between the inputs and outputs when training and applies it to the new data for validation. Other SoC models have been made in relation to specific batteries that take into account the different internal chemistries, the model presented in this research is able to generalize all the devices used, without taking into account this information.

Although the Conrad, Multipower and Yuasa batteries have similar chemistries, the Conrad trained model is also tested on a LiFePo battery and a hot water buffer. Results from Figures 19 and 20 prove that this model performs well with a different data type. The LiFePo performed well, as its data is similar to the other batteries. The explanation for the hot water buffer doing well is that even though the data is very limited and the validation data only included the discharge cycle of the buffer. It makes it is easier to estimate than a realistic scenario. The hot water buffer data does not charge, so there is no change in direction upwards in the y-axis, which is where most of the deviations occur in Figure 20. The discharge scenario plot also appeared linear in nature which makes it easier for the model to estimate the next step. The hot water buffer deviates most when it is almost fully discharged. The input data shown in Figure 2 of a battery charging has an starts to oscillate when being almost fully charged. This is what is happening with the hot water buffer, but for this case it is the temperature that is exponentially decreasing.

The retrained model performs worse than the initial model when tested with 3 batteries and a hot water buffer, except for the Multipower. This performance is due to the amount of data that the retrained model had for training. There is less data that the retrained model had available when it is being trained, which resulted in the results shown in in Figure 21. Although the second model had less data to use, it was still enough to achieve accurate estimations. For the Multipower, Conrad and LiFePo, there are spikes when the model peaked, which resembles the results of the ANN model and follow the trend of how the voltage data looks like. It is very sensitive to the change of direction of the SoC as seen at min 2800 and 3500 of the Multipower in Figure 21.

The hot water buffer results appear to be the poorest, with a slight offset of 15%. This may due to the low number of samples that it has, which is also seen in blue line of Figure 21. The expectation is that the hot water buffer would perform more accurately if it had more data. The hot water buffer appears to be more sensitive to the input values, as it oscillated throughout its plot, instead of having a clear line. The results would be much more accurate if it had more data to be trained with, as it would have more data to learn from and estimate the next step.

For the second retrained model as shown in Figure 22, the expectation is that the LiFePo would perform poorer, as it has a different battery chemistry. A reason for the better results would have to be in the amount of data of the LiFePo battery. There are visible peaks when the model changed direction in the Multipower and Yuasa results, which have been explained earlier as due to the input voltage shape. The hot water buffer performs much better compared to the previously retrained model, showing that these results are less sensitive to the input and are able to follow the result more closely. The hot water buffer results do not deviate when being almost fully discharged, this is due to the LiFePo data that is used for training the model. The LiFePo data used for training gives a battery usage scenario, so there is more that it can train from. Compared to the hot water buffer data, which is just a discharge scenario.

For the forecasted model, the results are less accurate that expected. The model is able to estimate when the battery is charging or discharging, but it very sensitive to the input data, which resulted in it having a 70.1% accuracy. The results from Figure 23 show that the model is able to determine when the battery is charging or discharging but the shape resembles the voltage input and not so much the measured SoC. An mentioned in Homans [8] research, this can be improved by predicting the voltage before estimating the SoC.

The forecasted model takes the SoC_{t-1} as the input for the model, the current and voltage data is known so it uses these values as a control parameter, so it is essentially estimating the SoC in real-time. This is different to how the other LSTM results are made, as the model gets run once and estimates based on that, while the forecasted version runs the model at every time step and only outputs one value.

Model Comparison

Although the ANN is a much simpler model than the LSTM it proved to achieve accurate results for two of the three batteries as shown in Table 1. The LSTM is able to achieve more accurate results compared to the ANN and is initially able to do so when being fitted with the Conrad data-set as shown in Tables 1 and 2. It is also shown that there is no offset in the results of the LSTM, as compared to the ANN results in Figure 18. Most likely due to the fact that an LSTM is a more complex model that is able learn better based on the inputs provided.

The ANN and LSTM use different methods. The ANN is a test case that only relied on separate charge and discharge data as training inputs, while the LSTM uses the scenario in Figure 6 as the input. Although the methods are different, the outputted accuracy can be compared.
Both models used the same data-set for validation. Figures 15-17 can be compared to Figure 18, where it can be seen that the estimation lines from the LSTM are much closer related to the actual values than the ones from the ANN (Figures 15-17). It can also be seen that Figures 18-22 do not show the exponential peaks as in the results from the ANN. This could be due to the higher complexity of the model and that the ANN is more sensitive to the changes in the data, which means that the ANNs SoC output shows the influence of voltage.

Finally, the model used a Conrad battery to forecast future results. The results in Figure 23 are 20% lower than the results without real-time estimation. The forecasts do not match the measured values. This is due to the fact that the forecast increases in errors the more steps it takes and gets further and further away from the desired values. Forecasting in smaller time steps would prove to show better results, as it will be able to handle the changes better.

This research used most of the same data used for the DiBu model [8], and the final results can be compared. The results of the DiBu model are shown in Appendix A, where the Conrad, Yuasa and Multipower batteries are shown. The results of the initially trained LSTM model and forecasted real-time LSTM model are plotted in Figure 24 below. The DiBu model performs better than real-time model presented in this research. Homans mode [8] used voltage prediction as a necessary step for SoC estimation. The predicted voltage and measured voltage follow the same pattern, but the prediction mitigates the sudden and exponential changes. After the prediction, the voltage shape resembles the shape of the measured SoC.

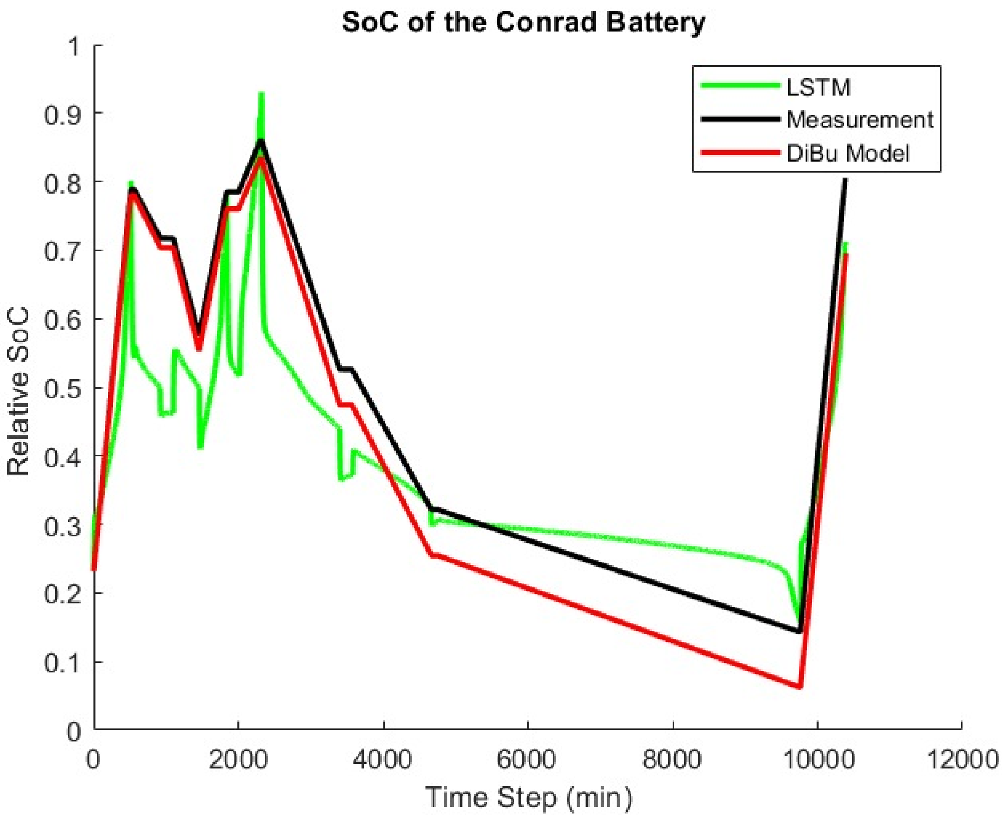


Figure 24: Results of LSTM and DiBu

5.2 Data set

The results are based on the data set. This data set is limited in many ways. The time duration for each battery is not large enough to see the full extent of how the model can behave, especially for the hot water buffer, where the data was very limited. Although there is limited data for the hot water buffer, it is key to test the model on a storage device that is not a battery. The battery data used is still enough to create a working model, but more data can result in a higher estimation accuracy as the model would have more data to learn from.

For the results shown in Tables 1-4 the MAE, MSE and RMSE have large variations, even when the accuracy is shown to almost be the same. This is due to the input data being in different ranges. For example, data can be recorded between 0-10 and 0-1000, a deviation of 5% will cause large errors in the 0-10 range and not in the 0-1000 range. This explains the high accuracy of 93.2% shown in Table

3 for the Multipower and lower accuracy of the hot water buffer with 69.3%. This can be fixed by normalizing the input data before it goes into the model.

The models are only tested using four batteries and a hot water buffer. The outcome of both models achieve a high accuracy in estimating the SoC, but the outcome is also limited to this particular data set. How would the models behave for the full duration of the battery life, or if the battery had zero degradation, or if the battery was at the end of its life. The current data is only measured during a short a period of time, having the full life of the battery may offer other insights. The issue with the battery being at the end of its life, is that is will increase the cumulative errors of the model. As battery does not behave the same way at the end of its life. Another possibility is to regularly retrain the model based on the most recent data, the parameters in this case will reveal the signs of aging. All these factors must be taken into consideration when evaluating the results of the model. If the model is only trained on a certain scenario, then it cannot offer accurate estimations for different scenarios that may adapt how the battery functions.

The outcome would also differ if more than three types of batteries are used, or if 100 batteries of the same type are used to train the model. This could have resulted in a different accuracy output of the model. Such that, the accuracy would only increase if the data quality of these 100 batteries are accurate. Training the model on more batteries will allow it to be better generalized.

With the limited data, the model is able to perform well. With more data, the model will learn more about the batteries past behaviour. The LSTM model would perhaps perform better due to the more complex data set and will be able to handle a larger input in data. The LSTM is a more complex model that has the capability to do so. The LSTM has the ability to find complex patterns and relationships in the data that was not initially discovered in the current data set.

6 Conclusion

The objective of the research is to investigate if a generalized machine learning model can be applied to estimate the SoC of energy storage devices. In this research an ANN and LSTM model were used.

6.1 Research Questions

The main research question that is formulated is:

- *What is the trade-off of applying machine learning for state of charge estimation of energy storage devices compared to applying existing complex models?*

Four sub-questions have been created in order to aid and evaluate the main questions.

What existing model/domain knowledge is already present and how do they compare?
The answer to first question is presented in Chapter 2, the literature survey. Where multiple SoC estimation models are compared for different types of energy storage devices and showed that many models are already present that are able to estimate the SoC. The models presented, included the impedance, model based and data driven methods. The Kibam [17] and Dibu model [8] offer insights into models that do not require many parameters but are able to yield accurate results. This is a difference to the Husnayin's method that requires elaborate calculations, computations and a large amount of data to yield an accurate result [16]. Machine learning models, such as an LSTM is able to generalize the method for SoC estimation, such that it can be used for different energy storage devices.

What measurable variables are important for the generalisation of the model, that are applicable to storage in general?
These variable exist in the electrical and thermal domains and can be used to generalise the model. The values are: current, voltage in the electrical domain and the flow and change in temperature in the thermal domain and the SoC. The current and voltage are the input for the ANN and the current, voltage and SoC_{t-1} are the inputs for the LSTM. With the SoC being the output for both models.

Which machine learning technique is the most applicable for estimation state of charge?
It is concluded in Section 2.5 that applying neural networks is the best option for creating a generalized model and for achieving accurate results. Two neural network models are created to test this theory. The rest of this theory is proven in the results in Chapter 4, where an ANN and an LSTM neural network have been used. The ANN is the most basic NN and the LSTM is a time series NN, the results are then compared. The LSTM is also used to forecast results and

demonstrate how it may react with real time inputs.

The first model is a simple ANN, to test the accuracy of a simple neural network, the second model, the LSTM, is superior to the ANN. Although the ANN model only provides accurate results for the Yuasa and Multipower battery, the LSTM model is generic and abstract and is able to achieve results that are close to the measured values for a majority of its tests. A high majority of the estimated results are closer to the measured values. Making it more reliable than the ANN, this can be shown in the accuracy of the two models. The LSTM yielded an accuracy of over 90% in Table 2 compared to the results of the ANN in Table 1. The LSTM model is able to provide accurate estimations when retraining the model using different batteries. This ultimately shows that the same model is able to be generalized and even be accurate training on a battery that is of a different chemistry (LiFePo), when the model was initially trained with the Conrad data.

Which evaluation metrics and methods allows for a comparative study?
Refers to Chapter 4, Tables 1-4, which presents the results in metrics. Four metrics are used, MAE, MSE, RMSE and accuracy. Results can differ based on which battery is chosen to train the model. The RMSE for example is prone to large errors, for example in Table 1. There is a difference between the 25% accuracy for the Conrad and 92.1% of the Yuasa, but the RMSE scores are quite close to each other at 9.12 and 6.59%, this is because at the errors when the battery is almost fully charged are really large. The RMSE resembles this difference, but the accuracy is very different.

The accuracy plays a big role in seeing how the model compared to the measured values and it can be easily seen which models performed best with this metric. The other metrics take into account the errors associated with the model and require more interpretation, these metrics are the MSE, RMSE and MAE. The accuracy is therefore used as the main method to interpret the results. The highest accuracy for the ANN shown in Table 1 is 92.1% for the Yuasa battery, while the LSTM model is able to produce accuracies of above 90% for all storage devices in the initial tests.

What is the trade-off of applying machine learning for state of charge estimation of energy storage devices compared to applying existing complex models?
This research has selected two different types storage systems, a battery and a hot water buffer. Two models are presented, the first one being an ANN, which is the most basic NN that can be used.

Out of three batteries in the ANN, it is able to have an accuracy over 90% for two batteries, which is an accurate result. The other two batteries sustained an accuracy of 83.3% and 25.7%, which is not great considering all the batteries are of the same type. Therefore the ANN model did not produce good results over all. The LSTM on the other hand was able to produce an accuracy of over 90% for the four batteries and one hot water buffer. This model is able to estimate two different types of batteries and a hot water buffer. The difference of the LSTM model compared to current methods is the complexity of the model. The LSTM did not require many input parameters and it used the previous SoC to estimate the next one. One of the issues with this model is transforming the data so that it could be put into the model.

A key part of this research is find out whether a generalized model is able to estimate different storage devices and yield a good accuracy. It is able to do based on on results shown in Section 4.2. The model is able to achieve a high estimation accuracy result with limited input parameters.

The DiBu model by Homan [8], is able to accurately estimate the SoC 'where the difference between the model and the state of charge calculated from measurements is less than 5%'. Figure 25 from Homan [8] compares its results to the Kibam model. The Appendix shows these results for the Conrad, Yuasa and Multipower batteries. When looking at Table 2, the accuracy that is obtained 96.4-97.7%, meaning a difference of 2.3-3.6% difference in measured and predicted values for the same three batteries. The LiFePo and hot water buffer had lower accuracies due to the limited samples and nature of the storage devices, therefore had accuracy errors of 5.6 and 7.5%. The real-time LSTM had a difference of 29.9% when compared to the measured and predicted values.

When comparing the models to the real-time model in Figure 23 from this research, the Dibu model performs much more accurately. The real-time forecasted model yields an accuracy of 70.1%, which is significantly lower than the DiBu model and the other models presented. As mentioned in the discussion, this is because the SoC output resembles the shape of the voltage instead of the target values.

The trade-off of using the model presented in this research compared to other models is that it is simple to use and is able to achieve highly accurate results for SoC estimation, as well as being able to be used on different energy storage devices. As mentioned in the previous paragraph, the DiBu model [8] is also able to achieve highly accurate results for a simple model. The biggest trade-off is that most other models are created with one energy storage device in mind and the model requires many complex calculations that increases the computational load. The models presented in this research can estimate the SoC quickly and sufficiently accurately. The data has to be adjusted so that it fits the 3-dimensional input requirements of the model. Due to the simplicity of the model, data only has to be inputted, and with minimal parameter changes, can be accurate.

6.2 Future Work

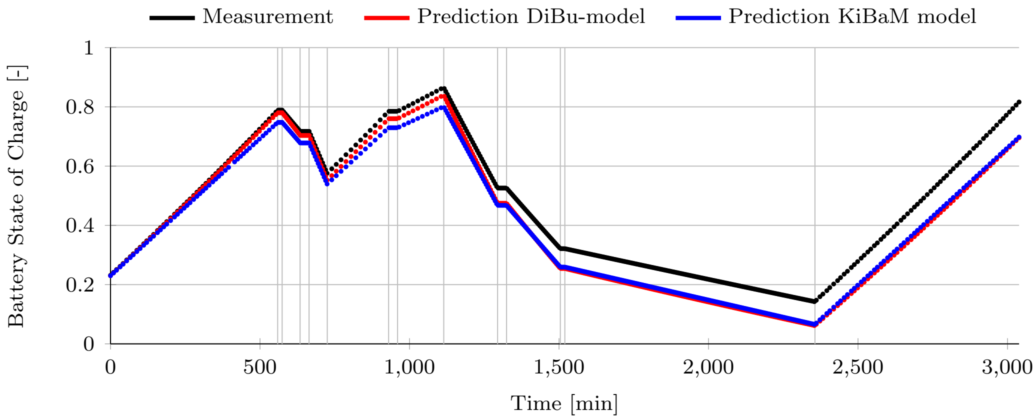
This section explains areas for future work. In order to find out how well the model works in a real scenario, the SoC should be estimated in real time, but unlike the way it is estimated in this research. The Dibu model [8] included voltage prediction as a necessary step, with this new voltage, the SoC real time estimation will be more accurate. Voltage prediction needs added in the next step of this research, to the initial LSTM model and the real-time version.

Creating a hybrid model such as, adding a Kalman or Particle filter, can achieve highly accurate results as presented in the literature Section 2.4. Hybrid methods combine two or more methods to estimate the SoC. Research into which extra model work well with an LSTM should be investigated.

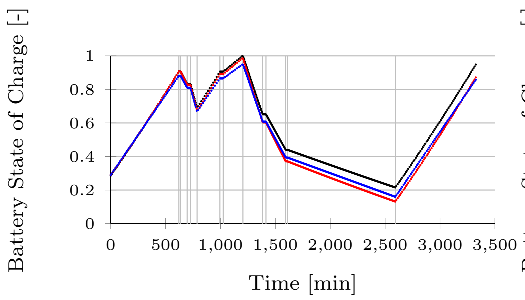
Another area that should be tested is that the model should be able to be used on energy storage devices in general. This assumes that the model should work on different storage devices and not just batteries and one data set from a hot water buffer. There should be more testing done into a variety of energy storage technologies. A creation of more different and elaborate storage systems should be investigated. Training the model with charging and discharging scenarios can improve the data quality. With the different scenarios, the model will be able to learn from different behaviours.

A Appendix

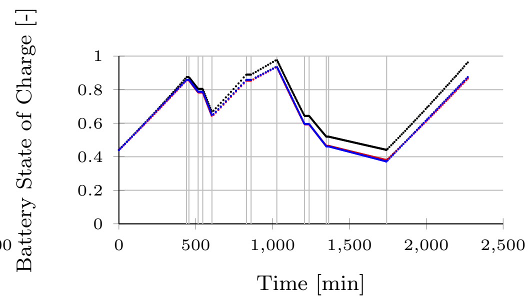
Figure taken from Homan [8].



(a) SoC of the *Conrad* battery.



(b) SoC of the *Yuasa* battery.



(c) SoC of the *Multipower* battery.

Figure 25: State of Charge of selected batteries, calculated from measurements on the batteries compared to predictions using the DiBu-model and the KiBaM model. The vertical lines represent the moment of step change.

References

[1] United nations framework convention on climate change. the paris agreement. [Online]. Available: https://unfccc.int,2015.Lastaccessedon11-6-2019

[2] Energiebalans; aanbod en verbruik. [Online]. Available: https://opendata.cbs.nl,2017.

[3] M. Meliani, A. Barkany, I. Abbassi, A. Darcherif, and M. Mahmoudi, "Energy management in the smart grid: State-of-the-art and future trends," *International Journal of Engineering Business Management*, vol. 13, 2021, https://doi.org/10.1177/1847979021103292.

[4] B. Mehta. The emerging need for artificial intelligence in energy storage. [Online]. Available: https://www.ionenergy.co/resources/blogs/artificial-intelligence-in-energy-storage-systems/

[5] MarketsandMarkets. Machine learning market worth 8.81 billion usd by 2022. [Online]. Available: https://www.prnewswire.com/in/news-releases/machine-learning-market-worth-881-billion-usd-by-2022-644444863.html

[6] L. Smyth. What is the role of machine learning in industry? [Online]. Available: https://www.engineerlive.com/content/what-role-machine-learning-industry

[7] H. Kazmi and J. Suykens, "Large-scale transfer learning for data-driven modelling of hot water systems," *IBPSA Building Simulation 2019*, 2019, doi:10.26868/25222708.2019.210352.

[8] B. Homan, "Batteries in smart microgrids," 2020, doi:10.3990/1.9789036549578.

[9] Bari, "Challenges in the smart grid applications: An overview," *International Journal of Distributed Sensor Networks*, vol. 4, pp. 1–11, 2014, https://doi.org/10.1155/2014/974682.

[10] H. Gong, E. Jones, A. Jakaria, A. Huque, A. Renjit, and D. Ionel, "Generalized energy storage model-in-the-loop suitable for energy star and cta-2045 control types," pp. 814–818, 2021, doi:10.1109/ECCE47101.2021.9595892.

[11] Z. Cai, G. Liu, and J. Luo, "Research state of charge estimation tactics of nickel-hydrogen battery," *International Symposium on Intelligence Information Processing and Trusted Computing*, 2010, doi:10.1109/IPTC.2010.91.

[12] W. L. T. Gao, "Machine learning toward advanced energy storage devices and systems," *iScience*, vol. 24, no. 1, pp. 1–33, 2021, https://doi.org/10.1016/j.isci.2020.101936.

[13] B. Homan, R. P. van Leeuwen, L. Zhu, J. B. de Wit, and G. J. M. Smit, "Validation of a predictive model for smart control of thermal and electrical energy storage," *2016 IEEE International Energy Conference (ENERGYCON)*, pp. 1–6, 2016, doi:10.1109/energycon.2016.7513872.

[14] C. Patyn, T. Peirelinck, G. Deconinck, A. Nowe, "Intelligent electric water heater control with varying state information," *SmartGridComm*, pp. 1–6, 2018, doi:10.1109/SmartGridComm.2018.8587453.

[15] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuska, & R. Belmans, "Reinforcement learning applied to an electric water heater: From theory to practice," *TRANSACTION ON SMART GRID 1*, 2015, https://doi.org/10.48550/arXiv.1512.00408.

[16] B. Homan, M. ten Kortenaar, J. Hurink, and G. Smit, "A realistic model for battery state of charge prediction in energy management simulation tools," *Energy*, vol. 171, pp. 205–217, 2019, https://doi.org/10.1016/j.energy.2018.12.134.

[17] J. Manwell and J.G.McGowan, "Lead acid battery storage model for hybrid energy systems," *Solar Energy*, vol. 50(5), pp. 399–405, 1993, doi: 10.1016/0038-092x(93)900602.

[18] C. Liao, "Battery state of charge estimation using neural networks," *Electrical Engineering Master Thesis*, 2019, https://doi.org/10.7916/d8-v0vk-g014.

[19] W. Chang, "The state of charge estimating methods for battery: A review," *International Scholarly Research Notices*, pp. 1–7, 2013, https://doi.org/10.1155/2013/953792.

[20] S. Prashant, S. T. Kok, I. M. Y. I. Bin, and M. Saad, "Overview of model-based online state-of-charge estimation using kalman filter family for lithium-ion batteries," *Renewable and Sustainable Energy Reviews*, vol. 113, 2019, doi:10.1016/j.rser.2019.06.04.

[21] R. P. van Leeuwen, I. Gebhardt, J. B. de Wit, and G. J. M. Smit, "A predictive model for smart control of a domestic heat pump and thermal storage," pp. 1–10, 2016, http://dx.doi.org/10.5220/0005762201360145.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84–90, 2017, https://doi.org/10.1145/3065386.

[23] C. Ephrem, "Intelligent state-of-charge and state-of-health estimation framework for li-ion batteries in electrified vehicles using deep learning techniques," *Electrical and Computer Engineering, McMaster University*, 2018.

[24] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad, and A. Ayob, "Neural network approach for estimating state of charge of lithium-ion battery using

backtracking search algorithm," *IEEE Access*, vol. 6, pp. 10 069–10 079, 2018, doi:10.1109/ACCESS.2018.2797976.

[25] Z. S. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics: A Journal of Theoretical and Applied Statistics*, vol. 182(1), 2003, doi:10.1080/02331880309257.

[26] M. Gao, Y. Liu, and Z. He, "Battery state of charge online estimation based on particle filter," *4th International Congress on Image and Signal Processing*, 2011, doi:10.1109/CISP.2011.6100603.

[27] X. Liu, C. Zheng, J. Wu, J. Meng, D.-I. Stroe, and J. Chen, "An improved state of charge and state of power estimation method based on genetic particle filter for lithium-ion batteries," *Energies*, vol. 13, no. 2, 2020, doi:10.3390/en13020478. [Online]. Available: https://www.mdpi.com/1996-1073/13/2/478

[28] T. Hansen and C.-J. Wang, "Support vector based battery state of charge estimator," *Journal of Power Sources*, vol. 141, no. 2, pp. 351 – 358, 2005, https://doi.org/10.1016/j.jpowsour.2004.09.020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775304010626

[29] E. D. B. Enache, "Estimating a battery SoC using neural networks," *International Symposium on Fundamentals of Electrical Engineering*, 2014, doi:10.1109/ISFEE.2014.7050636.

[30] E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach," *Journal of Power Sources*, vol. 400, pp. 242–255, Oct. 2018, doi:10.1016/j.jpowsour.2018.06.104.

[31] L. L. Y.M. Zhang, Y.B. Zhang, "Application of genetic algorithm and rbf neural network in network flow prediction," *IEEE International Conference on Computer Science and Information Technology*, vol. 3, p. 298–301, 2010, doi:10.1109/ICCSIT.2010.5564566.

[32] O. Linda, E. J. William, M. Huff, M. Manic, V. Gupta, J. Nance, H. Hess, F. Rufus, A. Thakker, and J. Govar, "Intelligent neural network implementation for soci development of li/cfx batteries," pp. 57–62, 2009, doi:10.1109/ISRCS.2009.5251344.

[33] W. Chang, "Estimation of the state of charge for a LFP battery using a hybrid method that combines a RBF neural network, an OLS algorithm and AGA," *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 603–611, 2013, https://doi.org/10.1016/j.ijepes.2013.05.038.

[34] M. Ismail, R. Dlyma, A. Elrakaybi, R. Ahmed, and S. Habibi, "Battery state of charge estimation using an artificial neural network," *2017 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 342–349, 2017, doi:10.1109/ITEC.2017.7993295.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* The MIT Press, 2016.

[36] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness

[37] S. Li, C. Ju, J. Li, R. Fang, Z. Tao, B. Li, and T. Zhang, "State-of-Charge Estimation of Lithium-Ion Batteries in the Battery Degradation Process Based on Recurrent Neural Network," *Energies*, vol. 14, 2021, https://doi.org/10.3390/en14020306.

[38] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018, doi:10.1109/TIE.2017.2787586.

[39] J. Chen, Y. Zhang, J. Wu, W. Cheng and Q. Zhu, "Soc estimation for lithium-ion battery using the lstm-rnn with extended input and constrained output," *Energy*, vol. 262, Part A, 2023, https://doi.org/10.1016/j.energy.2022.125375.

[40] E. Almaita, S. Alshkoor, E. Abdelsalam, and F. Almomani, "State of charge estimation for a group of lithium-ion batteries using long short-term memory neural network," *Journal of Energy Storage*, vol. 52, Part A, 2022, https://doi.org/10.1016/j.est.2022.104761.

[41] W. Shen, "State of available capacity estimation for lead-acid batteries in electric vehicles using neural network." *Energy Conversion and Management*, vol. 48(2), pp. 433–442, 2007, doi:101016/jenconman200606023.

[42] M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, and Z. W. Seh, "Predicting the state of charge and health of batteries using data-driven machine learning," vol. 2, 2020, https://doi.org/10.1038/s42256-020-0156-7.

[43] R. van Leeuwen, "Towards 100% renewable energy supply for urban areas and the role of smart control," *University of Twente*, 2017, doi:10.3990/1.9789036543460.

[44] T. H. R. Donato, "Machine learning systems based on xgboost and mlp neural network applied in satellite lithium-ion battery sets impedance estimation," *Advanced Computational Intelligence: An International Journal*, vol. 5, no. 1, pp. 01 – 20, January 2018, doi:10.5121/acii.2018.5101. [Online]. Available: http://airccse.org/journal/acii/current.html

[45] A. G. Kashkooli, H. Fathiannasab, Z. Mao, and Z. Chen, "Application of artificial intelligence to state-of-charge and state-of-health estimation of calendar-aged lithium-ion pouch cells," *Journal of The Electrochemical Society*, vol. 166, no. 4, pp. 605–615, 2019. doi:10.1149/2.0411904jes.

[46] W. Meng, Y. Min, L. Bo, W. Qiao, and X. Xinxin, "State of charge estimation of lithium-ion batteries using lstm and narx neural networks," *IEEE Access*, vol. 8, pp. 236–245, 2020, doi:10.1109/ACCESS.2020.3031340.

[47] M. Sanjeevi. Chapter 10.1: Deepnlp, lstm networks with math. [Online]. Available: https://medium.com/deep-math-machine-learning-ai

[48] Z. Chen, H. Zhao, X.Shu, Y. Zhang, J. Shen, and Y. Liu, "Synthetic stage of charge estimation for lithium-ion batteries based on long short-term memory network modeling and adaptive h-infinity filter," *Energy*, vol. 228, 2021, doi:10.1016/j.energy.2021.120630.

[49] Colah. Understanding lstm networks. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[50] Edureka. (2017) Recurrent neural networks, tensorflow tutorial. [Online]. Available: https://es.slideshare.net/EdurekaIN

[51] M. Chan, A. Witztum, and G. Valdes, "Integration of ai and machine learning in radiotherapy," *Sec. Medicine and Public Health*, 2019, https://doi.org/10.3389/frai.2020.577620.

[52] C. Huang, Z. Wang, Z. Zhao, L. Wang, C. S. Lai, and D. Wang, "Robustness evaluation of extended and unscented kalman filter for battery state of charge estimation," *IEEE Access*, vol. 6, pp. 617–628, 2018, doi:10.1109/ACCESS.2018.2833858.

[53] JJ. Mae and rmse metrics. [Online]. Available: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

[54] T. H. R. Donato, "Analysis of look back period for stock price prediction with rnn variants: A case study on banking sector of nepse," *Procedia Computer Science*, vol. 167, pp. 788–798, 2020, doi:10.5121/acii.2018.5101. [Online]. Available: http://airccse.org/journal/acii/current.html